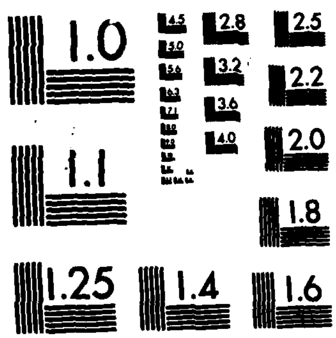
AD-A124 215 CEL-1 LIGHTING COMPUTER PROGRAM - PROGRAMMER'S GUIDE 1/3
(U) F AND K GROUP NEW YORK W E BRACKETT JAN 83
NCEL-CR-83.009 N68305-80-C-0012

UNCLASSIFIED F/G 9/2 NL

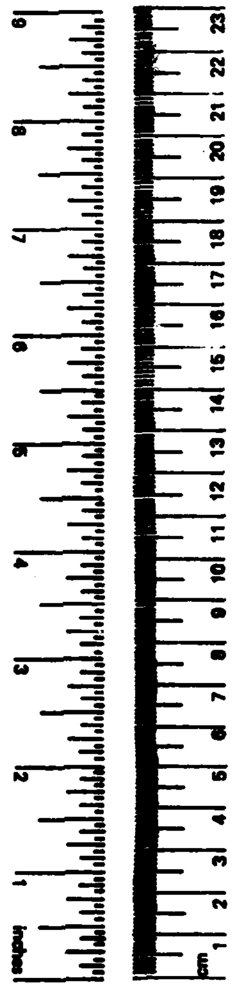


METRIC CONVERSION FACTORS

Approximate Conversions to Metric Measures

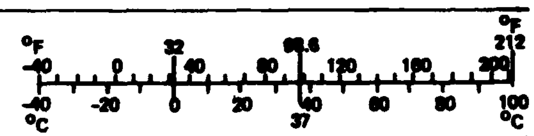
Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
in	inches	*2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
		AREA		
in^2	square inches	6.5	square centimeters	cm^2
ft^2	square feet	0.09	square meters	m^2
yd^2	square yards	0.8	square meters	m^2
mi^2	square miles	2.6	square kilometers	km^2
	acres	0.4	hectares	ha
		MASS (weight)		
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2,000 lb)	0.9	tonnes	t
		VOLUME		
tsp	teaspoons	5	milliliters	ml
Tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft^3	cubic feet	0.03	cubic meters	m^3
yd^3	cubic yards	0.76	cubic meters	m^3
		TEMPERATURE (exact)		
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

*1 in = 2.54 (exactly). For other exact conversions and more detailed tables, see NBS Misc. Publ. 286, Units of Weights and Measures, Price $2.25, SD Catalog No. C13.10:286.



Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
m	meters	1.1	yards	yd
km	kilometers	0.6	miles	mi
		AREA		
cm^2	square centimeters	0.16	square inches	in^2
m^2	square meters	1.2	square yards	yd^2
km^2	square kilometers	0.4	square miles	mi^2
ha	hectares (10,000 m^2)	2.5	acres	
		MASS (weight)		
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1,000 kg)	1.1	short tons	
		VOLUME		
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m^3	cubic meters	35	cubic feet	ft^3
m^3	cubic meters	1.3	cubic yards	yd^3
		TEMPERATURE (exact)		
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F



Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CR 83.009	2. GOVT ACCESSION NO. AD-A124 215	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) CEL-1 Lighting Computer Program - Programmer's Guide		5. TYPE OF REPORT & PERIOD COVERED Final Jan 1980 - Sep 1982
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) William E. Brackett, Applied Software Analysis, Boulder, CO 80301		8. CONTRACT OR GRANT NUMBER(s) N68305-80-C-0012
9. PERFORMING ORGANIZATION NAME AND ADDRESS The F + K Group, 475 Fifth Avenue, New York, NY 10017		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Z0362-01-211A
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Civil Engineering Laboratory, Port Hueneme, CA 93043		12. REPORT DATE January 1983
		13. NUMBER OF PAGES 213
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Lighting; daylighting; illumination; equivalent sphere illumination; visual comfort probability; luminance; energy conservation; lighting controls

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The basic algorithms and program file structure of the CEL-1 (Conservation of Electric Lighting, Version 1.0) lighting computer program are documented. The CEL-1 computer program aids the illumination engineer in designing energy efficient interior lighting systems. Lighting metrics which may be calculated include illuminance, luminance, equivalent sphere

DD FORM 1473, JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

illumination, and visual comfort probability. Energy profiles resulting from lighting controls which respond to daylight can be evaluated using CEL-1. This programmer's guide is divided into seven sections: (1) Programs Structure; (2) Basic Techniques; (3) Main Program Descriptions; (4) Subprogram Descriptions; (5) Logical Unit Assignments; (6) Compling the Programs; and (7) Source and Auxiliary Files.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd)

TABLE OF CONTENTS (Cont'd)

TABLE OF CONTENTS (Cont'd)

INTRODUCTION

This document details the algorithms and program file structure for the CEL-1 Lighting Computer Program. Before attempting to use this document the reader should be familiar with the CEL-1 LIGHTING COMPUTER PROGRAM User's Guide. This document also assumes the reader is well acquainted with lighting technology and terms.

The program descriptions given here and the commentary in the code itself are intended to be complementary. Where the program comments are sketchy, this document attempts to fill the void, etc.

Probably, the best way to use this document is to go over it in its entirety to grasp the nature of the overall structure and the basic techniques. It should then be considerably easier to delve into the details.

In any case the reader must understand the usage of these terms before he attempts to seriously get into any part of the document:

1. area points - the points on a control target area (used for dimming luminaires). See section 4.12 of the User's Guide.

2. direct component - means the illumination from luminaires (or daylight) only, before any reflections inside the room are considered.

3. FCT - read section 2.2 of this document.

4. fenestration source element - one window, one skylight, etc.

5. furniture - means a 6-sided rectangular object inside the room; used synonymously with the term "obstruction".

6. Illuminance at target point - in addition to illuminance, this term may also include target and background luminance at the point. Illuminance is synonymous with "illumination".

7. indirect component - means the total illuminance, less the direct component (i.e., the component due to reflections from interior surfaces.)

8. interior surface - can mean (1) one face of an obstruction, (2) a room surface, (3) a room surface insert, or (4) a fenestration rectangle.

9. luminance - used here in the old-fashioned sense. All reflecting surfaces are assumed Lambertian, which implies their luminous distribution. The "luminance", then, is a scalar which (in English units) may be thought of as a reflected footcandle (i.e., 1 fL = 1 lumen / square foot).

10. LUT - read section 2.1 of this document.

11. partition - almost always means one face of an obstruction within the room. Only rarely will it mean the thin-walled obstruction normally called a partition.

The reader is also urged to consult the references. In particular, RQQ No. 5 provides good general background for the fundamentals of interior point-by-point calculations, although it contains many formulae errors and misprints.

SECTION I

CEL-1 Programs Structure

1.1 Programs Structure

The CEL-1 package is divided into many separate programs. The division is necessary because of memory size limitations, but this division also enhances the package's modularity. Figure 1.1.a shows the CELPROC procedure file which is invoked to execute CEL-1; the program names are shaded. These are the programs which constitute the CEL-1 package proper. The package does contain a few additional auxiliary programs which are described later in this document.

Depending on the calculations required, certain of the CEL-1 programs are executed in a predetermined order. Key variables and accumulated illumination quantities are passed from one program to the next via temporary random access disc files which have the data stored in binary form. These files are frequently referred to simply as "binary files" in this document. The files are not saved after the CEL-1 execution terminates.

The programs may be thought of as divided into 3 subsets according to the CEL-1 capabilities which are being invoked. These three subsets are:

1. Electric Light subset
2. Daylight subset
3. Partition/synthesis subset

These subsets are discussed in the following sections.

Note that the appropriate subset for the desired calculations is selected automatically by the program code -- the user is never made aware of what particular subset has been invoked and hence never needs to worry about selecting a particular subset of programs. In the discussions and procedure file listings to follow, note that the NOS-provided internal switches SW<u>x</u> are used to execute particular subsets in the following way:

SW1 - Set in CEL01 to select the Electric Light subset.

SW2 - Set in CEL02 to select the Partition/synthesis subset.

SW3 - Set in CHECK if the input data deck contains errors; this bypasses all other programs.

If none of SW1, SW2, or SW3 is set, then the Daylight subset is selected by default.

1.2 Electric Light Subset of Programs

This subset of programs is invoked whenever the input deck has the following properties:

1. UNKNOWN task locations (rectangular grid)
2. No daylight
3. No design synthesis
4. No obstructions in the room

The Electric Light subset employs algorithms which may differ substantially from those used when furniture and/or daylight are present. The Electric Light subset algorithms are designed to efficiently compute illuminance at a large number of target locations.

The Electric Light subset consists of the hatched program names shown in Figure 1.2.a.

```
00070 COPY,INPUT,TAPE5.
00080 REWIND,TAPE5.
00090 REWIND,TAPE6.
00100 ATTACH,RELOC.
00105 LIBRARY,RELOC.
00108 GET,TAPE72=ERRORS.
00110 GET,CHECK.
00120 CHECK.
00122 IF(SW3=1)GOTO,00482.
00130 REWIND,TAPE5.
00135 REWIND,TAPE6.
00140 REWIND,OUTPUT.
00145 GET,TAPE64=RCS866.
00150 GET,CEL01.
00155 CEL01.
00160 IF(SW1=1)GOTO,00372.
00170 GET,CEL02.
00180 CEL02.
00190 IF(SW2=1)GOTO,00360.
00220 GET,CEL031.
00230 CEL031.
00240 GET,CEL032.
00250 CEL032.
00260 GET,CEL033.
00270 CEL033.
00272 GET,CELSOL.
00274 CELSOL.
00280 GET,CEL034.
00290 CEL034.
00300 GET,CEL035.
00310 CEL035.
00320 GET,SUNHIT.
00325 SUNHIT.
00340 GET,CEL07.
00345 CEL07.
00350 GOTO,00480.
00360,GET,CEL04.
00365 CEL04.
00370 GOTO,00460.
00372,GET,OVLY20.
00390 OVLY20.
00400 GET,OVLY30.
00410 OVLY30.
00420 GET,OVLY40.
00430 OVLY40.
00440 GET,OVLY50.
00450 OVLY50.
00460,GET,OVLY60.
00470 OVLY60.
00480,EXIT.
00482,REWIND,TAPE6.
00485 COPY,TAPE6,OUTPUT.
00490 PURGE,DAYFIL/NA.
00500 DAYFILE,DAYFIL.
00510 SAVE,DAYFIL.
```

Figure 1.2.a: The CELPROC procedure file. The Electric Light subset of programs is hatched.

1.3 Daylight Subset of Programs

The Daylight Subset of programs is used whenever daylighting calculations are called for. Furniture may or may not be present. The daylight subset is designed for efficiency in profile mode - i.e., it is geared toward performing daylight calculations for many different conditions, rather than only one instant during the year. The daylight subset is shown in the shaded portion of Figure 1.3.a.

```
00070 COPY,INPUT,TAPE5.
00080 REWIND,TAPE5.
00090 REWIND,TAPE6.
00100 ATTACH,RELOC.
00105 LIBRARY,RELOC.
00108 GET,TAPE72=ERRORS.
00110 GET,CHECK.
00120 CHECK.
00122 IF(SW3=1)GOTO,00482.
00130 REWIND,TAPE5.
00135 REWIND,TAPE6.
00140 REWIND,OUTPUT.
00145 GET,TAPE64=RCS866.
00150 GET,CEL01.
00155 CEL01.
00160 IF(SW1=1)GOTO,00372.
00170 GET,CEL02.
00180 CEL02.
00190 IF(SW2=1)GOTO,00360.
00220 GET,CEL031.
00230 CEL031.
00240 GET,CEL032.
00250 CEL032.
00260 GET,CEL033.
00270 CEL033.
00272 GET,CELSOL.
00274 CELSOL.
00280 GET,CEL034.
00290 CEL034.
00300 GET,CEL035.
00310 CEL035.
00320 GET,SUNHIT.
00325 SUNHIT.
00340 GET,CEL07.
00345 CEL07.
00350 GOTO,00480.
00360,GET,CEL04.
00365 CEL04.
00370 GOTO,00460.
00372,GET,OVLY20.
00390 OVLY20.
00400 GET,OVLY30.
00410 OVLY30.
00420 GET,OVLY40.
00430 OVLY40.
00440 GET,OVLY50.
00450 OVLY50.
00460,GET,OVLY60.
00470 OVLY60.
00480,EXIT.
00482,REWIND,TAPE6.
00485 COPY,TAPE6,OUTPUT.
00490 PURGE,DAYFIL/NA.
00500 DAYFILE,DAYFIL.
00510 SAVE,DAYFIL.
```

Figure 1.3.a: The CELPROC procedure file. The Daylight subset of programs is hatched.

1.4 Partition / Synthesis Subset of Programs

This subset is used whenever the design synthesizer is invoked and /or furniture is present in the room (no daylight allowed). The subset is hatched in Figure 1.4.a.

```
00070 COPY,INPUT,TAPE5.
00080 REWIND,TAPE5.
00090 REWIND,TAPE6.
00100 ATTACH,RELOC.
00105 LIBRARY,RELOC.
00108 GET,TAPE72=ERRORS.
00110 GET,CHECK.
00120 CHECK.
00122 IF(SW3=1)GOTO,00482.
00130 REWIND,TAPE5.
00135 REWIND,TAPE6.
00140 REWIND,OUTPUT.
00145 GET,TAPE64=RCS866.
00150 GET,CEL01.
00155 CEL01.
00160 IF(SW1=1)GOTO,00372.
00170 GET,CEL02.
00180 CEL02.
00190 IF(SW2=1)GOTO,00360.
00220 GET,CEL031.
00230 CEL031.
00240 GET,CEL032.
00250 CEL032.
00260 GET,CEL033.
00270 CEL033.
00272 GET,CELSOL.
00274 CELSOL.
00280 GET,CEL034.
00290 CEL034.
00300 GET,CEL035.
00310 CEL035.
00320 GET,SUNHIT.
00325 SUNHIT.
00340 GET,CEL07.
00345 CEL07.
00350 GOTO,00480.
00360,GET,CEL04.
00365 CEL04.
00370 GOTO,00460.
00372,GET,OVLY20.
00390 OVLY20.
00400 GET,OVLY30.
00410 OVLY30.
00420 GET,OVLY40.
00430 OVLY40.
00440 GET,OVLY50.
00450 OVLY50.
00460,GET,OVLY60.
00470 OVLY60.
00480,EXIT.
00482,REWIND,TAPE6.
00485 COPY,TAPE6,OUTPUT.
```

Figure 1.4.a: The CELPROC procedure file. The Partition/Synthesis subset of programs is hatched.

SECTION II

Basic Techniques

2.1 Asymmetric Lookup Table (Asymmetric LUT)

CEL-1 must frequently perform 2-dimensional interpolation to obtain a computed value. E.g., BRDF data is stored as a 2-dimensional array of factors on a hemispherical grid. A BRDF factor for a given azimuth and elevation is obtained via an interpolation in this array. Candela values for a given luminaire are usually available as a 2-dimensional array, etc.

The use of an "asymmetric lookup table" (frequently referred to hereafter as simply "LUT") is motivated by the benefits of avoiding trigonometry to the maximum extent possible, as trigonometric functions are computationally much more expensive than vanilla floating point arithmetic.

Suppose we have a point source luminaire mounted 10' above the target plane and we need to compute illumination at 100 target points on the target plane. The most straightforward way to accomplish this for each point is to determine the spherical angles (azimuth and elevation) which describe the target point location relative to the point source. These azimuth and elevation angles are used to access the candela table which characterizes the luminaire; a candela value is then interpolated from the 2-dimensional array of candela values. The candela value is then used in the inverse square law

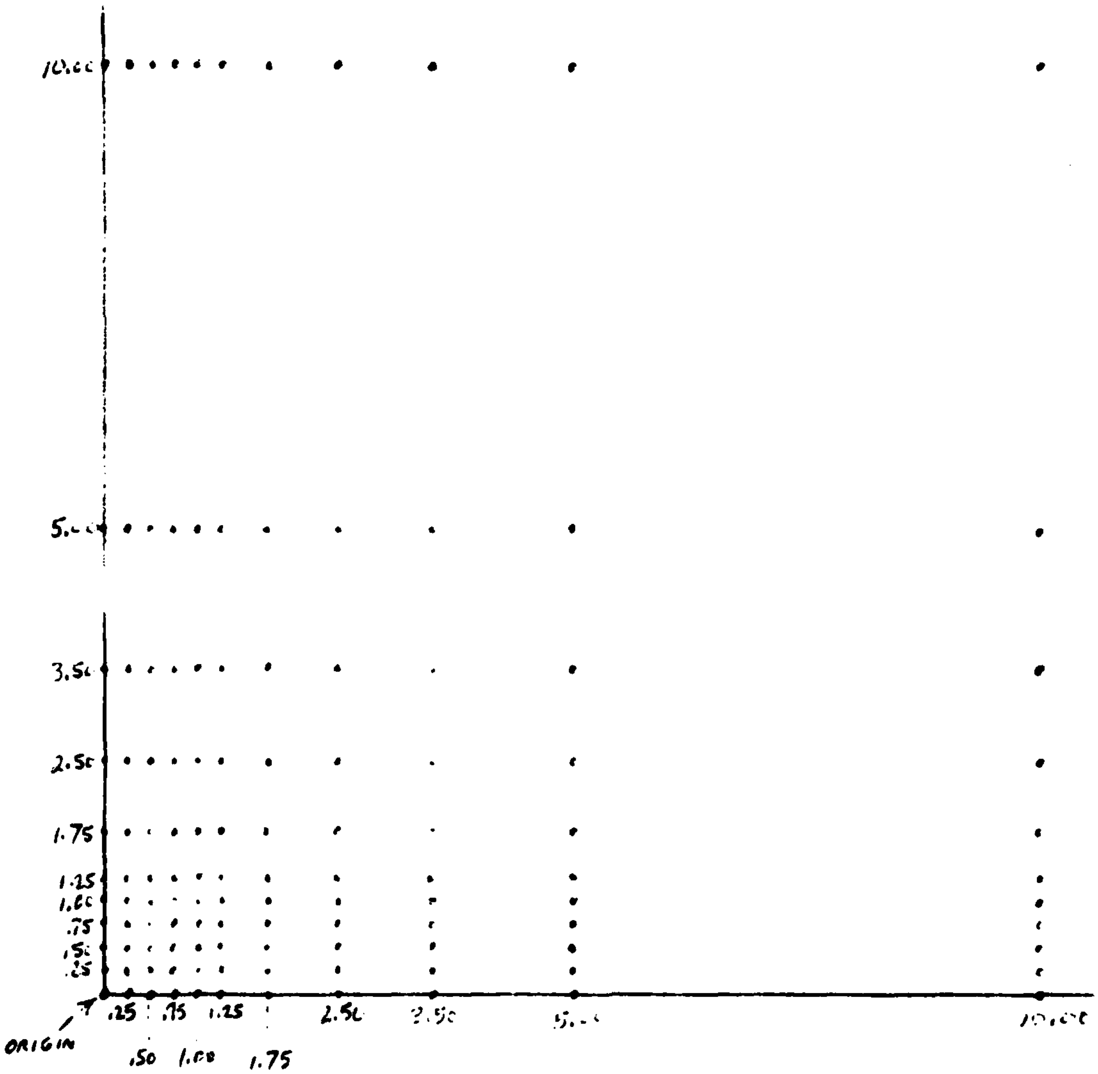
$$E = \frac{(\text{candela})(\cos\theta)}{d^2}$$

to compute the illumination at the target point. This process would be repeated for each other target point.

The painful part of the above procedure is that trigonometry (most conveniently, inverse tangent) must be employed to obtain the azimuth and elevation angular offset of the target point. Although the numbers in the example are small enough to make using trigonometry a "don't care" proposition, a look at a typical application makes it worthwhile to seek an alternative. First, luminaires are rarely effectively treated as point sources. This means that a luminaire must be treated as being composed of several point sources. Further, there are usually many luminaires present. Finally, point-by-point computations must be performed on the room surfaces as well as the target points.

Instead of applying the inverse square law in each case, we obtain a 2-dimensional grid of illumination values which spans all our target points. In that case, we could interpolate among the points in the grid which immediately surround the target point. No trigonometry is involved in such a process. The most convenient grid of this type to use would be one whose values were uniformly spaced in each direction. For a 10' mounting height we might need values every, say, 2½' in the vicinity of the luminaire to ensure ourselves of the desired accuracy. However, we will want the grid of values to span at least 50' (and possibly 100') on every side of the luminaire. A 100' x 100' grid at 2½' square spacing will require 41 x 41 = 1681 tabulated values; a 200' x 200' grid will require 4 times as many. Since a 2½' mesh seems like overkill in the remote regions of the grid (where values are small and gradients

Figure 2.1.a: Each dot represents a tabular value in the LUT. Interpolation is performed in the appropriate cell. The numbers give the offset of each row and column from the origin (offset is in mounting heights). Note that this figure shows only the upper right quadrant of the LUT; the pattern is repeated in each of the other three quadrants.



gentle), it seems sensible to use a coarser mesh far out and a finer mesh closer in to the luminaire. The asymmetric LUT is born!

(THIS SPACE INTENTIONALLY LEFT BLANK)

Refer to Figure 2.1.a. The point source luminaire is mounted 1' above the target plane. Using the inverse square law, illumination is computed at the grid points shown (in all 4 quadrants). This grid spans 10' on all sides and should have sufficient resolution everywhere to yield reasonable values when using 2-way linear (=hyperbolic) interpolation.

To see how this table is used in practice, suppose that the actual mounting height is 5' above the target plane and the target point is 10' east and 20' north of the luminaire. We scale the grid by a factor of 5 (the actual mounting height) and interpolate in the shaded region in Figure 3.1.b. After interpolation we must divide the value obtained by the square of the mounting height (in this case, 25).

The term "asymmetric LUT" arises from the fact that the grid spacing is not uniform. In the CEL-1 code, selection of the cell in which to interpolate is facilitated by the use of the array INDX. Since the finest spacing is ¼ mounting height, we can let INDX point to a particular cell based on the number of ¼-mounting heights the target point lies

from the lower left corner of the LUT. For example, suppose the mounting height is 10' and the target point lies 12' west of the luminaire. We must then interpolate in a cell which is (-1.25,-1.0) mounting heights from the luminaire. Since 12' is 4 x (88/10) = 35.2 ½-mounting heights from the west edge of the LUT, we must initialize the INDX array so that it points to the cell bounded by (-1.25,-1.0). In this manner we may, without a linear search through the cell boundaries, determine the LUT cell to interpolate in.

A second array ORD stores the ordinates of the LUT; this facilitates the interpolation also. We have ORD(1) = -10., ORD(2) = -5., ORD(3) = -3.5, ... , ORD(21) = 10. The following code section shows a typical use of an asymmetric LUT for interpolation (the function HYP is performing hyperbolic interpolation):

```
C
C LOOP THRU THE DISCRETE PIECES
C
      DO 92 JD=1,NDC
      XD = (XL/NDC) * (JD-0.5) - XL*0.5
      XX = (X-XD)/H
      XX = SIGN(1.,XX) * AMIN1(9.99,ABS(XX))
      LL = 4.* XX + 41.
      LL = INDX(LL)
      DL = (XX-ORD(LL))/(ORD(LL+1)-ORD(LL))
      DO 88 ID=1,NDR
      YD = (YL/NDR) * (ID-0.5) - YL*0.5
      YY = (Y-YD) / H
      YY = SIGN(1.,YY)*AMIN1(9.99,ABS(YY))
      II = 4.* YY + 41.
      II = INDX(II)
      DI = (YY-ORD(II))/(ORD(II+1)-ORD(II))
      E = HYP(FT(II,LL),FT(II,LL+1),FT(II+1,LL),
     <  FT(II+1,LL+1),DL,DI) * Q
      FC(I,J) = FC(I,J) + E
C
```

Asymmetric LUTs are pervasive throughout the CEL-1 code. One cannot hope to understand the code without mastering the LUT concept.

2.2 Fenestration Candela Table (FCT)

Since the profile mode requires the calculation of daylight effect for 45 different conditions (15 time instants multiplied by 3 sky conditions apiece - the simplicity of the overcast sky distribution actually reduces this to 31 separate computations), it is vital that the daylight effects be computed in as efficient a manner as possible. To this end the FCT is employed. The idea behind the FCT is to isolate all the static conditions affecting daylight calculations and compute as far as possible with these static conditions, leaving as little computation as possible for each of the different dynamic (i.e., sky luminance pattern) conditions.

Basically the approach boils down to computing the daylight effects assuming unit luminance outside the room. These values are stored away on disc and then read back in and adjusted for each different outdoor condition. The trick is in computing and storing the values so that they may be efficiently adjusted according to the complex luminance pattern prevailing outside.

Suppose that the room fenestration consists solely of a window small enough to be considered uniformly bright. Let us assume unit luminance outside the window; we can then compute the effect of the window (it is a Lambertian source of known luminance) on any target point; this illuminance is the "configuration factor"*. Once the configuration factor is known, for any given actual sky condition we need only know the luminance evident through the window; multiplying the precomputed value by the actual luminance yields the actual illuminance at the target point for each sky condition.

Getting this unit-luminance contribution is straightforward enough; the challenge lies in the method used to link the unit-luminance calculation with the actual sky brightnesses for the different sky conditions. The method used in CEL-1 is to always calculate the sky luminance at a fixed set of sky positions and to express the precomputed unit-luminance contribution as a linear combination of the sky luminances at these fixed sky positions.

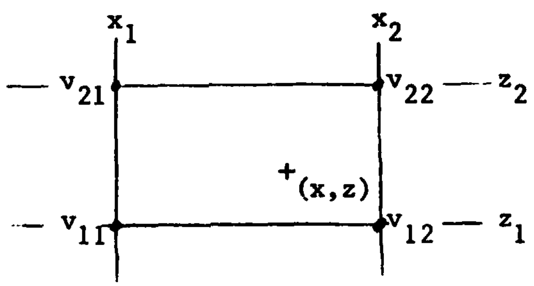
These fixed sky positions are the destinations of a 21 x 21 array of rays projected outward from and normal to the window in question. The angular separations of these rays are such the rays form an asymmetric LUT (see section 2.1). In this case the LUT "origin" is the center of the window and the LUT calculation points are the sky points determined by the rays projected at the LUT angles.

* The configuration factor is the illuminance at a given point due to a given Lambertian (i.e., perfectly-diffusing) area source of unit luminance. Configuration factors are computed (in somewhat disguised form) in the subroutine STABLS.

The "link" between the unit-luminance (configuration factor) calculation and the sky brightness calculation is this: The actual luminance evident through the window piece will be obtained by interpolating among 4 of the sky points for which brightness will be calculated. Thus, the problem boils down to locating the target point's position within the LUT of sky brightnesses and then weighting the configuration factor among the 4 corners of the LUT cell we must interpolate in. This process is detailed in the following paragraphs.

To determine the sky luminance we construct an LUT whose origin is at the center of the window and where the "mounting height" is a distance perpendicular to the window and extending into the room. If we then project a ray from each LUT grid point through the center of the window, we get a mesh of 441 rays projected outside the window. Using sky luminance distribution formulae and the luminance of ground and other buildings, we can associate a luminance with each LUT grid point whenever the sky conditions are specified.

We determine the target point's location in the LUT by using as "mounting height" the distance from the target point to the wall the window lies on. In general, we will have isolated a cell within the LUT. In the sketch below, + represents the position of the target point within the 4 LUT nodes which immediately surround it:



Recall that without knowing anything of the daylight conditions we may compute the illuminance at the target point due to the window (provided we assume unit luminance; the computed value is the configuration factor) -- call this value H. Then for a

specified sky condition, the actual illuminance at the target point is

$$E = LH$$

where L is the luminance of the sky evident through the window. When the actual sky conditions are specified we will know the exterior luminance values at each LUT point, in particular the values v_{11}, v_{12}, v_{21}, and v_{22} as shown in the sketch. The luminance L is:

$$L = D_{11}v_{11} + D_{12}v_{12} + D_{21}v_{21} + D_{22}v_{22}$$

where

$$D_{11} = \frac{(x_2-x)(z_2-z)}{(x_2-x_1)(z_2-z_1)} \qquad D_{21} = \frac{(x_2-x)(z-z_1)}{(x_2-x_1)(z_2-z_1)}$$

$$D_{12} = \frac{(x-x_1)(z_2-z)}{(x_2-x_1)(z_2-z_1)} \qquad D_{22} = \frac{(x-x_1)(z-z_1)}{(x_2-x_1)(z_2-z_1)}$$

The D_{ij} are simply the hyperbolic interpolation coefficients; the crucial point is that they depend solely on geometry and may thus be precomputed with no knowledge of the actual sky condition. We would then save the D_{ij}, retrieve them for each different sky condition, and thus compute L each time (what is saved is actually $D_{ij}H$, H for unit luminance).

Thus far we have assumed the window is small enough that the luminance evident outside it may be considered constant. IN general this will not be the case. We will discretize the window so that each piece may be considered as revealing constant luminance outside. For a given target point we then determine 4 D_{ij} multipliers for each window piece. In general, many LUT cells will be spanned by this process; in the worst case, each and every one of the 20x20 cells in the LUT would be involved.

The algorithm may be stated as follows:

1. Purpose is to calculate a table of 441 multipliers M_{ij} for i=1,...,21 and j=1, ... ,21. These multipliers correspond one for one with LUT grid points. They will ultimately be multiplied by exterior luminance values L_{ij} to obtain illumination E at the target point:

$$E = \sum_{i=1}^{21} \sum_{j=1}^{21} M_{ij} L_{ij}$$

2. Set $M_{ij} = 0$ for all i and j

3. Take a window piece small enough that the exterior luminance evident through it can be considered constant.

4. Locate the LUT offset of the target point from the window. Let the cell be isolated by the i^{th} cell vertically and the j^{th} cell horizontally.

5. Compute the hyperbolic interpolation factors D_{ij}, $D_{i+1,j}$, $D_{i,j+1}$, and $D_{i+1,j+1}$

6. Set $M_{ij} = M_{ij} + D_{ij}H$

$$M_{i+1,j} = M_{i+1,j} + D_{i+1,j}H$$

$$M_{i,j+1} = M_{i,j+1} + D_{i,j+1}H$$

$$M_{i+1,j+1} = M_{i+1,j+1} + D_{i+1,j+1}H$$

where H is the illuminance at the target point due to the window, assuming unit luminance outside.

7. Repeat steps 3-6 until each discrete window piece (including other windows on the same wall) has been enumerated.

The table of M_{ij} values thus computed is called the FCT for the target point in question.

The following comments apply to the use of the FCT:

1. For each target point, a separate FCT must be computed for each different fenestration type. E.g., a window on the south all implies a different FCT than one on the west wall.

2. The factor H is computed as indicated in step 6 of the algorithm, when the calculation point is on the target plane. When computing illumination on an interior surface a different technique is used. Here the window piece is treated as a point candela source whose contribution is given by:

$$E = \frac{L}{\pi} (A \cos\theta \cos\beta) \;/\; d^2$$

where L = luminance evident through the window

θ = angle between normal to the target surface and the vector from the point to the window piece

β = angle between the normal to the window and the vector from the point to the window piece

A = area of the window piece

d = distance from the target point to the window piece

In this case the factor H is given by $H = \dfrac{A \cos\theta \cos\beta}{d^2}$

3. While each target point - fenestration type combination results in one FCT, this is not the case with illumination on an interior surface. We are interested in the illuminance on interior surfaces only as a component in the flux transfer analysis to determine the final average luminance. Therefore we are interested in the average (not point-by-point) initial illumination on each interior surface.
Since we need only the average illumination on an interior surface, we need only one FCT per interior surface. If n is the number of points on the interior surface, then step 6 of the algorithm is modified to loop thru all n points and the computation is changed to:

$$M_{ij} = M_{ij} + \frac{1}{n} D_{ij} H \quad \text{etc.}$$

The resulting FCT gives the average initial illuminance (due to the fenestration in question) over the interior surface.

4. It is important to realize that each FCT is computed only once. On the other hand the luminance values at the LUT points must be computed for each different sky condition under consideration. The calculation of these luminance values requires a substantial amount of ray-tracing and trigonometry just to determine the destination of the rays projected thru the LUT points outside the room. Again these calculations need not be repeated for each different sky condition. Therefore for each fenestration type the following are precomputed for each of the 441 LUT points (projected rays):

a) The destination of the ray (sky, ground, ground insert, other building surface).
b) For rays which strike the sky: azimuth and elevation of the point on the sky.
c) For rays which do not strike the sky:
   i. A number identifying the surface struck.
   ii. The (x,y,z) coordinates of the point struck.
   iii. The cosine of the angle between the ray and a normal to the surface.

The collection of the above data for each of the 441 LUT ray destinations is called a

<u>Prototype FCT</u>

2.3 Obstructions

The terms "obstruction" and "furniture" are used synonymously in this document to mean 6-sided rectangular objects located in the room interior. The term "partition", while it may infrequently be used to mean a piece of furniture with one very small dimension, is used generally to mean on and only one surface of a piece of furniture.

Calculations involving obstructions imply two primary capabilities:

1. Ability to account for the direct shadowing of partitions.

2. Reckoning of the indirect component to account for the partition. E.g., if we change nothing in the space except the reflectance of one partition, its shadowing effect has not been altered. Nonetheless, the illuminance at the target plane will have changed, and we want to be able to measure the change.

2.3.1 List Structure

A fundamental data structure in handling obstructions is the linked list. A linked list is a method of storing array-oriented data so that an element may be relocated without exhanging it with another element. This is particularly handy when an "element" consists of not one but several characteristics -- this is precisely the case with partitions.

First we "explode" each obstruction into its six component faces, each of which we call a partition. After we have done this, we add the room surfaces (six additional partitions) and any inserts and fenestration rectangles. The result is a collection of planar interior surfaces, each of which may be referred to as a partition. We use the following FORTRAN arrays to characterize each partition:

$$C1 \begin{cases} X1 - \text{the x-coordinate nearest the origin} \\ Y1 - \text{the y-coordinate nearest the origin} \\ Z1 - \text{the z-coordinate nearest the origin} \end{cases}$$

$$C2 \begin{cases} X2 - \text{the x-coordinate furthest from the origin} \\ Y2 - \text{the y-coordinate furthest from the origin} \\ Z2 - \text{the z-coordinate furthest from the origin} \end{cases}$$

NROWS - # rows of points on the surface to be used in calculating initial illuminance point-by-point on the surface.
NCOLS - # columns of points on the surface

DIR - the direction the surface faces (1=west, 2=north, etc.)

REFL - reflectance of the surface

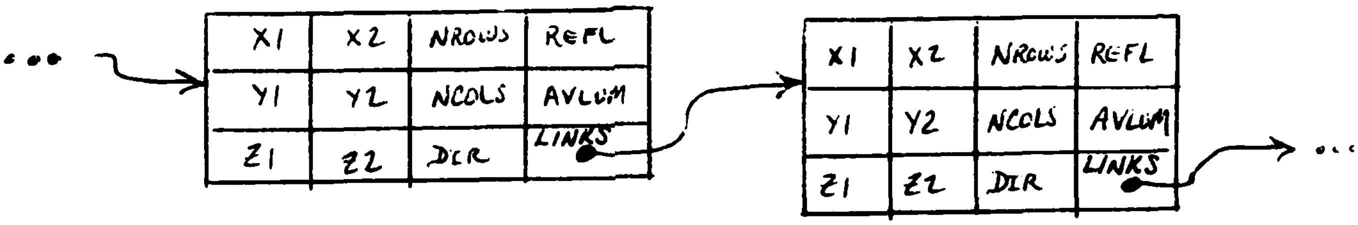
AVLUM - average luminance on the surface

The notation above means that C1 is a 3-column array whose first column is X1, second column is Y1, and third column is Z1. Ditto for C2.

Each list cell may be thought of as one element from each of these arrays; visually each cell can be thought of as a cluster of fields:

X1	X2	NROWS	REFL
Y1	Y2	NCOLS	AVLUM
Z1	Z2	DIR	LINKS

To construct a list, it remains only to link cells together with the array called LINKS. For example, the cells could be linked as in



The virtue of this scheme is that link pointers can be rearranged without physically moving the array elements X1, X2, ... around in storage. For example, additional link fields may be added to join only those cells with a given property; for example, a separate list might be constructed of only those cells facing west. This is in fact done in the CEL-1 code. The 6-element vector HEAD contains 6 pointers, one to each list facing one of the 6 directions. This arrangement constitutes the basic list structure from which all other lists are constructed. The LINKS array is used to join the cells in this direction-oriented list structure.

2.3.2 Accounting for Obstructions in Direct Component Calculations

The shadowing effect of partitions is reckoned by a ray-tracing scheme which determines if any partitions can block the target point's view of the luminous source. A ray-trace involves the calculation of the vector between the source and target; it is then determined whether or not this ray passes through a given partition surface. The expense of the calculation varies with the number of partitions for which the check is made; therefore it is worthwhile to look at as few partitions as possible.

Two separate schemes are used to prune the list of partitions to be searched:

1. Illumination calculations on target points:

In this case a series of four quadrant-oriented partition lists is constructed. When each luminous source (piece of luminaire or fenestration) is considered, four separate lists are constructed, one for each quadrant surrounding the luminaire piece. The quadrants are numbered as follows, where + denotes the luminaire or fenestration piece:

```
      !
   2  !  1
      !
 -----+-----
      !
   3  !  4
      !
```

The partitions which belong in the list for quadrant 1 are those which could interfere with target points in that quadrant. Note that this excludes all partitions possessing any of the following properties:

a) No part of the partition lies in the quadrant
b) The partition faces east
c) The partition faces north

Analogous properties exclude partitions from the other quadrants.

2. Illumination calculations on interior surfaces:

In this case a target surface may span more than one quadrant, so that a quadrant-oriented partition search is not appropriate. Instead, an imaginary plane through the luminous source is constructed. A single list of potentially interfering partitions is then constructed, which includes all partitions except those possessing either of the following properties:

a) No part of the partition lies between the imaginary plane and the target plane
b) The partition faces the opposite direction as the target plane

2.3.3 Accounting for Obstructions in Indirect Component Calculations

The indirect component calculation requires two basic steps:

1. Determine the final luminance on each interior surface.
2. Using the final luminances from 1, determine the effect on all target points.

With no obstructions in the room, the determination of the final luminances is reasonably straightforward: the traditional flux transfer analysis is performed, using the initial illuminances on each

room surface (direct component from luminaires and daylight) and the form factors* (which are computed from known formulae). The problem with obstructions is that the form factor formulae presume that no obstructions are present.

The compromise which CEL-1 employs is to compute the form factors between surface i and surface j as if no intervening surfaces were present. For a given surface i, all such form factors F_{ij} are computed, for all other surfaces $j \neq i$. In general, the sum of these F_{ij} will be greater than 1 whenever obstructions are present, whereas reality dictates that the sum must be exactly 1. Hence each of these F_{ij} is proportioned downward so that the sum of the adjusted F_{ij} is exactly 1. The flux transfer solution giving the final luminance on each surface is then determined in the customary way.

Once these final luminances have been determined, it remains to compute the effect at each target point. For each target point this is done by projecting 48 rays into the upper hemisphere at predetermined azimuth and elevation angles. Each of these 48 rays is traced through the partitions until it strikes one of these interior surfaces. The product of the luminance of the surface struck and a precomputed multiplier associated with the ray in question gives the illuminance at the target point from the zone surrounding that particular ray. The sum of all 48 such products gives the indirect component at the target point.

It is important to note that for all interior surfaces which are walls or ceiling, the final average luminance is considered constant over the entire surface. For walls and ceiling, however, each surface is divided into as many as 400 zones, where each zone has its own unique final luminance and is considered a "surface" itself for the 48-ray projection scheme.

It is also worth noting that future enhancements may consider the resolution with 48 rays too coarse, so that the scheme applies directly to a larger number of projected rays.

* The "form factor" from surface i to surface j is the fraction of reflected flux leaving surface i which is incident upon surface j. This factor depends only upon the geometry of the environment and in no way upon the magnitude of the surface brightnesses. For more details and the algorithms used, refer to the documentation for functions FFPER and FFPAR.

2.4 Effect of Sunlight on Target Points

The effect of the sun on target points inside the room involves (1) the direct solar contribution and (2) the reflected solar contribution.

The direct solar contribution is the illuminance at a point which is exposed to direct sunlight. The illuminance due to direct sunlight is a straightforward calculation (subroutine HFCS). The problem is to determine whether or not the sun is visible from a point in the room. This determination is made by a ray-tracing scheme which sees if the ray from the sun to the target point is obstructed in any way by any solid surfaces such as furniture in the room, external buildings, etc.

The reflected solar contribution is that illuminance at target points which is due to interreflections among sunlit interior surfaces. To compute the brightness on the interior surfaces due to direct sunlight, ray-tracing is done which is similar to that performed for the direct solar component.

The user should note that the direct sun component is computed only for horizontal illumination (i.e., not for ESI); the reflected solar component is computed for both horizontal illuminance and for ESI.

SECTION III

Main Program Descriptions

3.1 CHECK

Purpose

CHECK examines the user's input data deck for errors and inconsistencies. If any errors are detected, a diagnostic message is generated and execution of the remaining programs should be aborted. If no errors are detected, execution of the remaining programs in the stream can proceed.

Method

The easiest way to see what error conditions are detected by CHECK is to scrutinize the auxiliary file ERRORS which contains the diagnostic messages which identify errors. In general, CHECK is block-oriented -- i.e., it reads one input block and checks it for errors. If any errors are found, a diagnostic is generated and the remainder of the input in that block is ignored. The input stream is read until the next block keyword is encountered, and error checking can resume. In this manner, errors may be detected in more than one block.

When an error is detected, the corresponding diagnostic from ERRORS is printed, along with the last input line read. Each line in ERRORS may be up to 80 characters long. The first four characters are reserved for a 3-digit number followed by one space. The 3-digit number is not used by CHECK, but its presence in the ERRORS file helps the user pinpoint each diagnostic when modifying the file.

Subroutines called by CHECK perform the following functions:

DISECT
This routine reads an input line and counts the number of variables on the line. It returns what type each variable is: integer, real, or alphanumeric.

GFORM
Returns the FORMAT which may be used to print an input line with the minimum number of non-blanks.

ICH
This function isolates the I^{th} character (I = 1 to 10) in the input word W. The character code is returned right-justified in the function result word ICH; the 9 leftmost character positions in W are set to zeroes.

FLUSH
This routine reads the input stream until a block keyword is encountered. It is used immediately after an error has been detected.

FORM

FORM uses GFORM to return the FORMAT which prints an error diagnostic with a minimum number of trailing blanks.

ER

This routine prints the error diagnostic indicated by the parameter M.

OBDB

OBDB sets the parameter NDBID to the number of obstruction database entries.

CCDB

CCDB sets the parameter NCCDB to the number of cloudiness database entries.

3.2 CEL01

Purpose

CEL01 is used in all subsets of the CEL-1 package. It reads the entire input data deck and creates the binary disc files which facilitate subsequent interprogram communication.

Method

CEL01 performs the following functions:

1. Writes the input data echo to the print file.
2. Constructs all binary disc files which can be generated from the input stream only.
3. If input units are in meters, converts them to feet.
4. "Explodes" obstructions into the list of interior partition surfaces.
5. Generates the following precomputed tables:
   a) BRDF factor LUTs (subroutine BRDF)
   b) Zonal multipliers for 48-ray projections (subroutine ZONEPC)
   c) Wall and ceiling contribution LUTs (subroutine STABLS)

Subroutine SCOMP is used to match character strings from the input deck to the block and sub-block keywords. It tells the main program when a new block of input data is beginning.

3.3 CEL02

Purpose

CEL02 computes and saves the contribution (both direct + reflected components) from each luminaire to the target points, area points, and interior sensors.

Method

1. The needed binary disc files are read into memory.

2. The flux transfer matrix is computed and written to disc (subroutine WRFTM).

3. If we have a rectangular grid of target points (UNKNOWN task locations), the grid is converted to a linear list.

4. The 48-ray projections from each target point are generated and their destinations saved on disc (subroutine D048). Note that this step must be performed in daylight environments even when no luminaires are present.

5. For each luminaire, the direct and indirect components are computed before the next luminaire is considered. The following steps are performed:

   a) A point source LUT is constructed if the luminaire photometry or orientation is different from that of the previous luminaire.

   b) The luminaire contribution is accumulated on all interior surfaces (subroutine ACCAS).

   c) Quadrant-wise partition lists are built around the luminaire (subroutine CPARTL).

   d) For each target point, the luminaire is discretized and its direct component computed using the point source LUT.

   e) Step d) is repeated for both area points and for interior sensors.

   f) Flux transfer analysis yields the final average luminance on each interior surface (subroutine FTAPAR).

   g) The point-by-point luminances on the walls and ceiling are computed. (subroutine ADJFL)

   h) The interreflected component at the target points, area points, and interior sensors is computed.

6. The results are written to disc.

3.4 CEL031

Purpose

CEL031 computes the FCTs for the interior surfaces, area points, and interior sensors. The routine also computes the relative initial illuminances (due to fenestration) point-by-point over the walls and ceiling.

Method

1. The necessary binary disc files are read in from disc.

2. The fenestration map entries are constructed (subroutine CONMAP).

3. If this is an energy profile calculation and blinds angle settings have not been specified, the settings are computed (subroutine VFBLA).

4. The FCTs are computed (subroutine FCTDSK).

5. The relative initial illuminances on walls and ceiling are calculated (subroutine RSDET).

3.5 CEL032

Purpose

CEL032 generates the FCTs for the target points.

Method

The required binary disc files are read in, and subroutine FTPDSK is called to compute the required FCTs.

3.6 CEL033

Purpose

CEL033 computes the prototype FCTs. The prototype FCTs contain the destinations of the rays projected outside the room along fenestration LUT angles.

Method

The required binary disc files are read in. For each fenestration map entry, subroutine PROFCT is called to compute the desired prototype FCT.

3.7 CEL034 (also subroutine TRIGS)

Purpose

CEL034 computes the daylight effects for all the different time instants required and writes these to disc for subsequent processing.

Method

1. The necessary binary disc files are read into memory.

2. 48 rays are projected from each exterior sensor location.

3. If we're in energy profile mode, the time instants for which to perform calculations are determined (subroutine CTIMES).

4. The effect of an overcast sky with fixed zenith luminance (= 1000 fL) is calculated and saved.

5. For each time instant the luminance of the overcast sky is computed and the results from 4. are factored according to the ratio of zenith luminances to produce the actual overcast sky effect at each time. Note that only one call to DAYEF is required -- this is a consequence of the simplicity of the overcast sky distribution.

6. For each of the clear and partly cloudy skies, DAYEF is called for each time instant to compute the daylight effect.

The subroutine TRIGS computes trigonometric functions of solar azimuth and elevation.

3.8 CEL035

Purpose

CEL035 prints the results of daylight calculations, for either profile or analysis mode.

Method

1. The required binary disc files are read into memory.

2. The array (INDW) is written which associates each luminaire with quadratic coefficients yielding watts vs. gain.

3. The first and last hours of building occupancy are determined.

4. The cloudiness database is read in.

5. If luminaires are to be controlled individually, the LGRP array must be adjusted so that the # of dimming groups equals the number of luminaires.

6. CMPRES is called to minimize disc accesses required during the profile calculations.

7. All daylight is "extinguished" and the corresponding luminaire gain settings are determined. The resulting energy consumption is that which will prevail before sunrise and after sunset.

8. For each time instant, gain settings are determined (subroutine MSTRSN). Results of each calculation (daylight + luminaires) are printed. In profile mode, only a statistical summary of the target point values is printed. For analysis mode, this step is the final step.

9. In profile mode, we now have the following:

   For each of the three days Dec. 22, March 21, and June 21 we have the watts consumption for each of 5 time instants each day. Further, at each time we have these figures for each of the three sky conditions.

   We first wish to obtain, for each sky condition of each of the three days, the consumption at each occupied hour during the day. We proceed as follows:

   a) For a given day and sky condition, we have known consumption vs. time of day for 5 different times. Call OSCU to fit an osculating polynomial over these data.

   b) For each occupied hour, obtain the consumption from the interpolating coefficients computed in a).

c) Repeat a) and b) for each other day and each other sky condition.

10. After step 9, we have the watts consumed at each occupied hour for each sky condition for each of the three days. We treat the consumptions on March 21 as applying equally to September 21. The consumption figures for a typical day each month are then computed via linear interpolation between two days for which values are available.

    E.g., to get the consumption at 2pm on November 15 (clear sky), we interpolate between the 2pm clear sky figures for Sept. 21 and Dec. 22.

11. PROFIL is called to print the results.

Variables / Arrays

WUSE - watts consumption for each of the 15 time instants and 3 sky conditions.

EN - contains the output of step 9. EN(I,J,K) = consumption at day I, hour J, sky condition K.

I = 1	Dec. 22
2	Mar. 21
3	June 21
4	Sep. 21
5	Dec. 22
6	Mar. 21

ENERG - the energy profile consumption. ENERG(I,J,K) = consumption at day I, hour J, sky condition K. The 12 I values are for the middle of each month.

3.9 SUNHIT (also subroutines BARR3, RAYSFW, ADJFP, IRASUN, CSUN, QUART, RANGE)

Purpose

In a daylighting environment, SUNHIT prints a character plot of the area on the target plane which is exposed to direct sunlight at any time during the year.

Method

The program determines a mathematical representation of the continuous sky region which the sun traverses during the year. From each point on the target plane which corresponds to a position on the character plot, several rays are projected through each fenestration source element. If any one of these rays strikes the continuous region of solar traverse, that point on the target plane is marked a "hit".

The sky region of sun traversal is the region which is bounded by the sun's paths on the longest and shortest days of the year. A quartic polynomial (elevation vs. azimuth) is used to represent the solar position on both the shortest and longest days of the year.
Each fenestration source element is taken in turn.

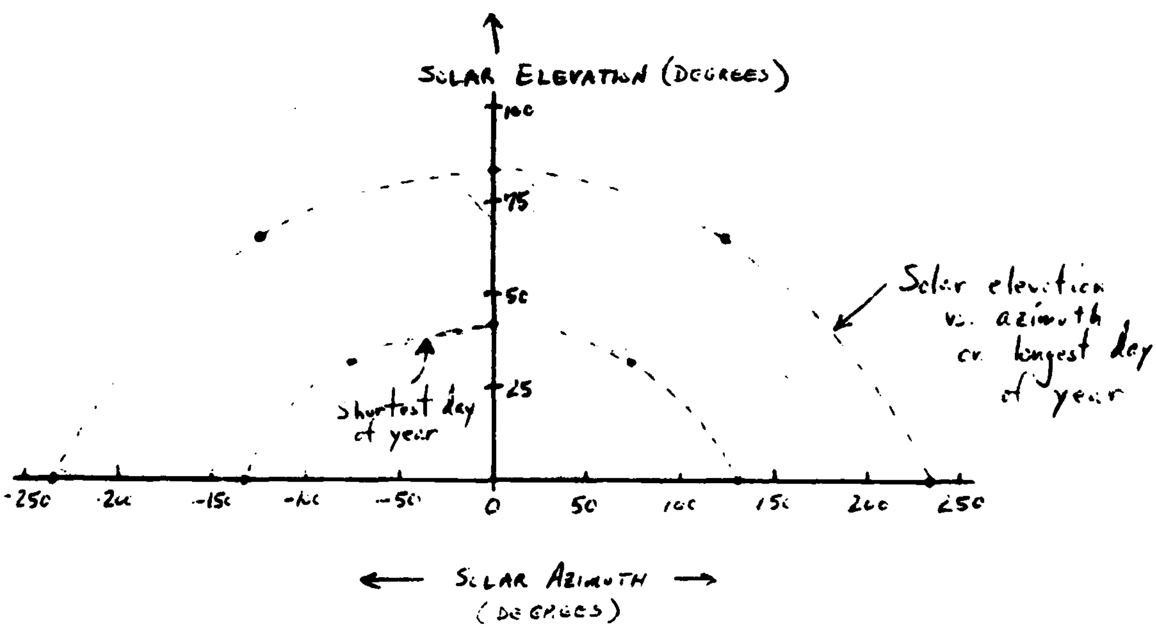


Figure 3.9.a: The shaded region is a typical region of solar traversal.

Subroutine BARR3

This routine constructs a barrier list. The list consists of fr... 0 to 3 sets of coordinate limits which locate any non-vanishing window barriers.

B1 - (x,y,z) coordinates of the barrier corner nearest the origin
B2 - (x,y,z) coordinates of the barrier corner furthest from the origin
KBCON - identifies the barrier's constant coordinate (1=x, 2=y, 3=z)

Subroutine TPLOOP

This routine takes each character position in the room plot to (x,y) coordinates in the room. RAYSFW is called for each such point and the array element corresponding to the plot point is set:

$$KSUN(I,J) = \begin{cases} 1 & \text{sun is not visible} \\ 2 & \text{sun is visible} \end{cases}$$

Subroutine RAYSFW

This routine determines whether or not sunlight hits a target point. The fenestration rectangle and target point locations are given. Rays are projected from the target point along the perimeter of the fenestration rectangle at intervals not exceeding 2'. E.g., a 2' x 2' or smaller window would have exactly 4 rays projected through it. If barriers are present, each projected ray which strikes a barrier is adjusted till it passes at the outer edge of the barrier (subroutine ADJFP).

The ray projection logic must first see if the fenestration point is above the ceiling. If so, the projected ray must not intersect the ceiling on its way to the point. Second, the ray must be traced to see if it strikes any exterior building surfaces (RASTRK). If these tests are passed, IRASUN is called to make the final determination.

Function IRASUN

IRASUN traces a ray from a plot point to a fenestration perimeter; the function determines whether an interior partition surface obscures the view of the fenestration point from the plot point. If not, the azimuth and elevation of the ray are computed, and it is determined whether the ray strikes the region of sun traversal.

Subroutine CSUN

Quartic coefficients are determined which give the solar elevation as a function of azimuth. Arrays are set as follows:

DAWN - solar azimuth at dawn each day (longest and shortest)

DUSK - solar azimuth at dusk each day (=-DAWN)
ENOON - solar altitude at solar noon
C - quartic coefficients of elevation vs. azimuth

Note: 1. All times are solar times.
2. On the shortest day above the arctic circle, the sun never rises. Specific adjustment is required for this.

Subroutine QUART

This routine computes 5 quartic coefficients; hence 5 points are required (solar elevation vs. solar azimuth):

azimuth	elevation	time
A1	0	sunrise
A2	Z	sunrise + .4 (noon - sunrise)
0	ENOON	noon
-A1	Z	noon + .6 (noon- sunrise)
-A2	0	sunset

3.10 CEL07

Purpose

CEL07 prints the BLAST interface information which constitutes a partial BLAST input deck. The intent is to free the user from having to input the same parameters to both BLAST and CEL-1, and also to print out the lighting schedules computed by the energy profile algorithm.

Method

1. The lighting schedules are printed out in BLAST-compatible form.

2. Wall locations, window locations, etc., are converted to the coordinates expected by BLAST and are printed out.

3.11 CEL04

Purpose

CEL04 is used in design synthesis or in a non-daylight partitioned environment. In the former case, CEL04 directs the design synthesis procedure; in the latter case, the results are read from disc and simply accumulated and printed (no optimization).

Method

1. The required files are read in from disc.

2. Luminaire contributions are compressed in order to minimize disc accesses (subroutine CMPRES).

3. Optimum operating set is computed (subroutine OPTMZ).

4. Results are printed (subroutine RSLTS). Note that dummy arguments are supplied where RSLTS expects daylight parameters.

3.12 OVLY20 (also subroutines OVLY21, OVLY22, OVLY23, OVLY24, AREALT, ACCUM, INDIR, ACCBS, VCPTBL, and ACCVCP)

Purpose

OVLY20 accumulates the direct contribution from luminaires to the target points and room surfaces.

Method

1. Required disc files are read into memory.

2. NDIV is adjusted so that point-by-point room surface illuminance will be calculated at the corners of the discrete room zones, rather than at their centers.

3. The contribution of each luminaire is accumulated as follows:

   a) if the luminaire photometrics or orientation differ from that of the preceding luminaire, a new point source LUT is calculated (subroutine OVLY22).

   b) The contribution of the luminaire to the 4 walls is accumulated.

   c) If the orientation or the mounting height or the photometrics for this luminaire are different from the previous luminaire, then:

      i) new area source LUTs are computed (one for the floor, one for the target plane, one for the ceiling). This is done in subroutine OVLY23.

      ii) VCP LUTs are computed (subroutine OVLY24).

   d) If the luminaire has significant downlight,

      i) the area source LUT for the floor is used to accumulate the contribution to the floor (subroutine ACCUM).

      ii) the area source LUT for the target plane is used to accumulate the contribution to the target points (subroutine ACCUM).

   e) If the luminaire has significant uplight,

      i) the luminaire contribution to the ceiling is accumulated (subroutine ACCUM).

      ii) the contribution of the "bright spot" above the luminaire is accumulated on the target points (subroutine ACCBS).

      iii) the ceiling luminance from the "bright spot" is saved, since it must be subtracted from the final average ceiling luminance before the interreflected component due to the latter is computed.

f) If VCP is being calculated, these quantities are accumulated:

n	# sources in field of view
$\sum L_i \omega_i$	luminance times solid angle
$\sum L_i Q_i / P_i$	luminance x Q function / position index
$\sum \omega_i$	solid angle subtended by luminaires

Subroutine OVLY21

OVLY21 reads the photometric data (subroutine PHOTO). If the required photometric file does not exist in the user's catalog OVLY21 attempts to read it from the master catalog M4800GS.

Subroutine OVLY22

OVLY22 call CLUT (see section 4.21) to compute the point source LUT. Also, ZETA is set to the angular separation between room nadir and photometric nadir.

Subroutine OVLY23

OVLY23 computes area source LUTs. An area source LUT is a lookup table whose origin is an area source luminaire. Consequently, the target plane must be a fixed distance away, since the $1/d^2$ correction factor (which may be used to apply point source LUTs over any mounting distance) does not apply to area sources. Three sets of area source LUTs are computed:

1) floor (horiz fc only)
2) target plane (horiz fc, raw fc w/body shadow, background luminance, task luminance)
3) ceiling (horiz fc only)

Note that 1) and 2) are computed only if the luminaire has significant downlight; 3) is computed only if the luminaire has significant uplight.

OVLY23 also computes ceiling luminance on the "bright spot" above the fixture (subroutine INDIR).

Subroutine AREALT

This routine computes one or more LUTs for an area source luminaire. If the indicator ILUM is zero, only one LUT (horiz fc) is computed. If ILUM=1, then in addition to the horizontal fc LUT, 3 more are computed for each viewing direction (one apiece for raw fc w/body shadow, background luminance, and task luminance).

Each LUT point is taken in turn, the luminaire is discretized according to the parameter THRESH, and the contribution of each taken from the point source LUT. Other variables / arrays are:

VIEW - the viewing directions
FT - point source LUT
SH - table of body shadow factors
BB - table of background luminance BRDF factors
BT - table of task luminance BRDF factors
FC - output LUT for horiz fc
RFC - output LUT for raw fc w/body shadow
BL - output LUT for background luminance
TL - output LUT for task luminance

Subroutine INDIR

This routine projects a grid of square zones above an uplight luminaire, then computes the average initial luminance on each square due to the one luminaire only. These squares constitute the "bright spot" above the fixture; treating each square as a "luminaire" accounts for the "first bounce" contribution from the actual luminaire to the target points.

The size of each square zone is 0.54 H, where H is the distance from the luminaire to the ceiling. The number of zones (subject to a maximum grid size 10 x 10) is chosen so that at least 2 zones in each direction are not above any part of the luminaire. See Figure 3.12.a.

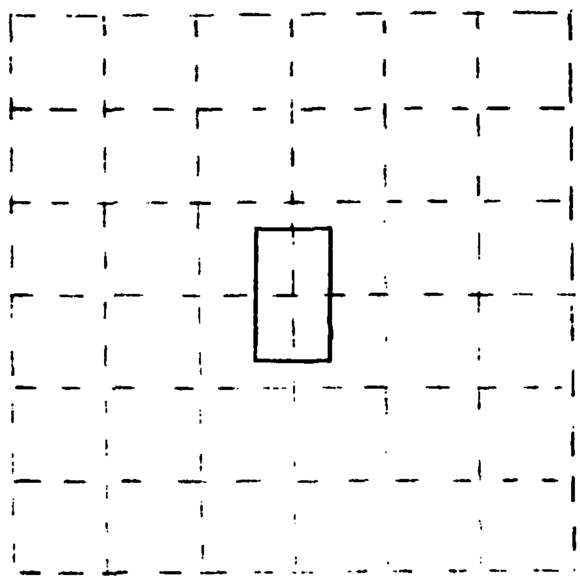


Figure 3.12.a: The dashed squares are the zones which collectively constitute the "bright spot" above the luminaire. INDIR determines the placement of the square zones and computes the average luminance on each square. The rectangle with solid edges represents the luminaire.

The average luminance in each square is computed by symmetrically placing a 4x4 grid of points in the square, using the area source LUT for fc on the ceiling to compute the illuminance at each of the 16 points, then averaging these results and multiplying by the reflectance.

HL - height of luminaire upper luminous surface above the floor
HC - height of ceiling
LEW,LNS - x and y dimensions of luminaire
FAF - area source LUT for fc on ceiling
S,B - interpolation coefficients (see section 4.76)
AL - average luminances on the zone squares
AI - work space for computing the 4x4 grids on each square zone
U - ½ the x-span of all zones together
V - ½ the y-span of all zones together
NEWZ - # zones running East-West (x-direction)
NNSZ - # zones running North-South (y-direction)

Subroutine OVLY24

This routine computes LUTs to be used in accumulating VCP intermediate quantities. The point source LUT for horizontal fc on the floor is read in, VCPTBL is called to compute the LUTs, and the LUTs are then written to disc.

Subroutine VCPTBL

VCPTBL computes LUTs of the following quantities[5]:

OMEGA - ω_i = solid angle subtended by the luminaires

LOM - $L_i \omega_i$ = luminance x solid angle

LQP - $L_i Q_i / P_i$ = luminance x Q function / position index

The following points should be noted:

1. ω_i for each luminaire is its projected area at each LUT point, divided by the square of its distance from the point.

2. L_i is given by $L_i = \frac{\text{(candela)}}{A \cos\theta}$

where $A \cos\theta$ = projected area of luminaire

candela is obtained from the horizontal fc LUT:

$$\text{candela} = \frac{fc\ d^2}{z^2 \cos\theta}$$

where fc is the value from the LUT
d is the distance from luminaire to target point
z is the mounting height of the luminaire above the target pt
θ is the angle between the normal to the ceiling and the line from the luminaire to the target point

3. P (position index) is computed according to Levin[6].

4. Tabulated values of ω_i, $L_i\omega_i$, $L_i\omega_i/P_i$ are computed at the four compass directions. If the actual viewing directions are different from these, linear interpolation between compass direction values is used to obtain the LUT values.

Subroutine ACCUM

ACCUM accumulates interpolated values from an LUT into a result array.

LUR - logical unit # of results disc file
LU12 - logical unit # of disc file where LUTs are stored
LUT - record offset for reading LUTs from LU12
IOFS - record offset for writing results to LUR
CON - factor applied to each value computed from the LUT
IDO - 4 x 4 table which enumerates the result arrays to be computed. Worst case is 13 (horizontal fc + 3 tables apiece (raw fc w/ body shadow, background luminance, task luminance) for each of four viewing directions.

Subroutine ACCBS

This routine computes the contribution of a "bright spot" above an uplight luminaire. First, (M,MM) and (N,NN) are set to indicate the zone boundaries whose centers lie within the room. Figure 3.12.b shows a bright spot near the northwest corner of a room:

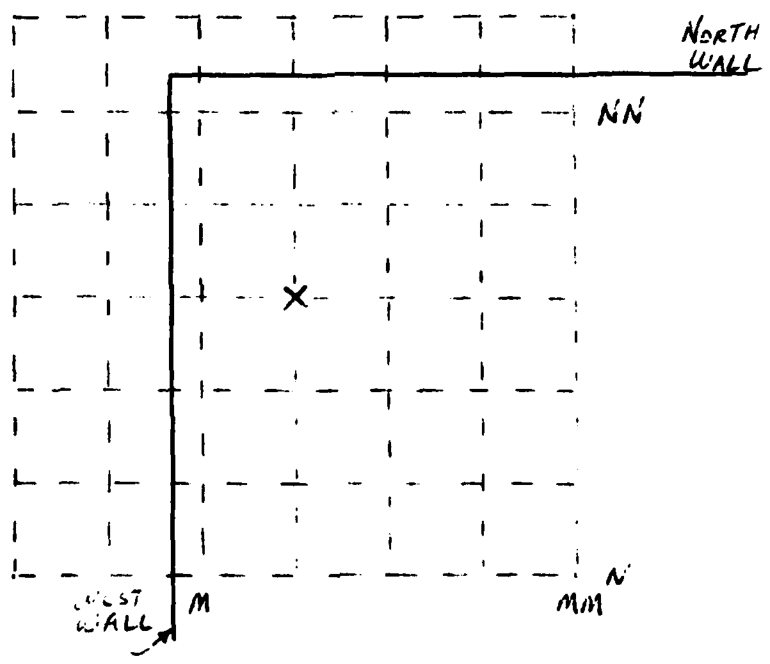
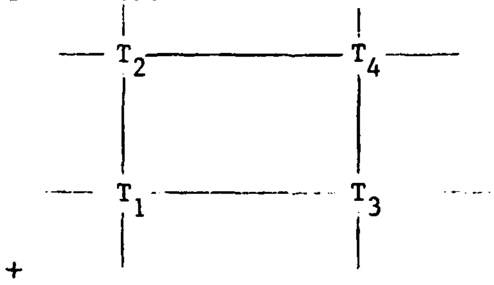


Figure 3.12.b: Only the contribution from the shaded zones of the bright spot above the luminaire are taken into account. The X represents the center of the luminaire.

Only the contributions of those zones whose boundaries are delimited by (M,MM) and (N,NN) are to have their contribution added. The contribution of each zone is obtained from the luminance of the zone (AL) and the LUT values at each corner of the zone (the LUTs are the ceiling contribution tables computed in subroutine STABLS). For example, let + represent the target point in the sketch below; the T_i represent the LUT values at each corner of the zone.



The contribution of the zone is (luminance of zone) x $(T_1-T_2-T_3+T_4)$

Subroutine ACCVCP

This routine accumulates the VCP LUTs to the target points. The following quantities are accumulated:

$$\omega_i$$

$$L_i\omega_i$$

$$L_iQ_i/P_i$$

n (# luminaires visible)

Variables / Arrays are:

A - contains LUTs as they are read in from disc
B - accumulates results
EN - contains ω_i value computed from LUT
NV - number of viewing directions
NR,NC - number of rows, columns of target points
XS - x-coordinates of each column of target points
KS - LUT indices corresponding to each XS
TORG - (x,y,z) coordinates of target point nearest origin
DELR,DELC - row, column spacing of target grid
XL,YL,ZL - (x,y,z) coordinates of luminaire
VIEW - viewing directions
XEW,YNS - x,y dimensions of luminaire
LU26 - logical unit # of disc file where LUTs are stored
LU27 - logical unit # of disc file where accumulated results are stored

3.13 OVLY30 (also subroutines ILAVG, RFAVG, FLUM, PERFF, INVRS)

Purpose

OVLY30 performs the flux-transfer analysis to obtain the final average luminance on each room surface. All the work is done by the following subroutines:

Subroutine ILAVG

ILAVG computes the average initial illuminance on each room surface. It does this by taking the weighted average of all point-by-point initial illuminance on the surface. Weights are necessary because the point-by-point calculations include the boundaries of the surface. Boundary points must be given weight = 0.5; corner points must be given weight = 0.25.

Subroutine RFAVG

This subroutine computes the average reflectance on each room surface. Each point on a discretization zone corner is assigned the reflectance of the surface, unless the point falls on an insert. In the latter case the point is assigned the reflectance of the insert. The weighted average reflectance of all such points becomes the average reflectance for the surface. The weights are chosen in exactly the same fashion as in subroutine ILAVG.

Subroutine FLUM

FLUM performs a 6 x 6 flux transfer analysis to determine the final average luminance on each room surface. See Section 4.94 for a discussion of flux transfer analysis. The analysis is simpler here than in 4.94, because we have only 6 surfaces, and never any obstructions here.

The form factors F_{ij} are written into the array A; the room surface areas and reflectances are factored in to obtain the flux transfer matrix B. B is inverted and final luminance obtained from

$$FLUM = -B^{-1}\rho\, E_0$$

where ρ is the vector of reflectances and E_0 is the vector of initial illuminances.

Note that since we have only to determine the form factors from each whole room surface to each other surface, we only need a means of computing form factors between perpendicular surfaces; the remaining form factors follow from the requirement that

$$\sum_j F_{ij} \quad \text{for any room surface } i$$

For example, suppose we compute

$$F_{12} \quad \text{(west wall} \dashrightarrow \text{north wall)}$$

and $$F_{15} \quad \text{(west wall} \dashrightarrow \text{floor)}$$

Then

$$F_{14} = F_{12} \quad \text{(west wall} \dashrightarrow \text{south wall)}$$

$$F_{16} = F_{15} \quad \text{(west wall} \dashrightarrow \text{ceiling)}$$

and $$F_{13} = 1 - 2(F_{12} + F_{15}) \quad \text{(west wall} \dashrightarrow \text{east wall)}$$

Function PERFF

This function computes the form factor from one room surface to an adjacent perpendicular surface. The method is the same as in section 4.44, except the formulation is simplified here because the source and receiving surfaces always share a boundary.

Subroutine INVRS

INVRS computes the inverse of matrix A via an analog of the Gauss-Jordan method for solving a system of linear equations. In Gauss-Jordan elimination, the right-hand side vector is appended as an additional column to the matrix of coefficients. Row-reduction operations are then performed which ultimately reduce the coefficients matrix to an identity matrix. If these same row-reduction operations are performed on the appended right-hand-side vector, then at the end of the process the appended column will contain the solution vector.

The same principle is employed in INVRS, except that an identity matrix B is appended to A. Each row reduction operation on A is likewise performed on B. Thus, when A has been reduced to the identity matrix, B will contain A^{-1}.

3.14 OVLY40 (also subroutines ENORM, EPARL, ADWAL, ADCEIL)

Purpose

OVLY40 computes the interreflected component on the target points. The final average point-by-point surface luminances are also computed.

1. The final point-by-point illuminances on each room surface due to each of the other room surfaces are computed. The LUTs computed in subroutine STABLS are used for this -- the horizontal fc LUT for wall contribution is used for perpendicular surfaces; the horizontal fc LUT for ceiling contribution is used for parallel surfaces.

2. The point-by-point final luminances on each wall are used to compute the average luminance above the target plane. This average above the target plane is used to compute the point-by-point effect at the target points due to the wall (subroutine ADWAL).

3. The average ceiling luminance is adjusted to account for ceiling luminance used in the "bright spot" effect calculations (OVLY20). The adjusted average is used to compute the effect of the ceiling on the target points.

Note that the ceiling contribution due to the "bright spots" generated in OVLY20 may be considered the "first bounce" effect. The ceiling contribution computed here in OVLY40 may be considered the "later bounces". The reason for the split is that any gradient on the target plane due to the interreflected component should be due to the first bounce; therefore the first bounce deserves more careful modeling than the later bounces.

Subroutine ADWAL

ADWAL accumulates the contribution of one wall to the target points. The wall contribution LUTs (computed in subroutine STABLS) are used.

IW - wall # (1, 2, 3, or 4)
TORG - (x,y,z) coordinates of the target pt which is nearest the origin
NR,NC - # rows, columns of target points
DELR,DELC - row, column spacing of target grid
ROOMD - the x, y, and z room dimensions
FC - accumulated contribution at the target points
T - holds the wall contribution LUT
RLUM - luminance of the wall

Subroutine ENORM

This subroutine computes the illuminance point-by-point at a discretized surface on the x-y plane due to an adjacent surface in the y-z plane. Via an appropriate axis rotation in the calling program, ENORM can be used to compute the contribution from any room surface to any adjacent surface.

Subroutine EPARL

EPARL computes the illuminance point-by-point at a discretized surface in the x-y plane from a parallel plane of the same size directly above it. Via an appropriate axis rotation in the calling program, ENORM can be used to compute the contribution from any room surface to any adjacent surface.

Subroutine ADCEIL

ADCEIL accumulates the contribution from the ceiling to the target points. The ceiling contribution LUTs (from subroutine STABLS) are used.

TORG - (x,y,z) coord
TORG, NR,NC,DELR,DELC, ROOMD, FC - these are the same as in subroutine ADWAL
T - holds the ceiling contribution LUT
RLUM - luminance of the ceiling
S,B - interpolation coefficients (see section 4.76)

3.15 OVLY50 (also subroutines CALVCP, INTSRF)

Purpose

OVLY50 prints the results which have been previously accumulated.

Method

1. If input units are metric, all distance coordinates are converted from feet back to meters.

2. Each indicator (INDC(I), for I= 1 to 10) is considered and a quantity printed out whenever INDC(I) = 1.

3. If INDC(14) = 1, room surface luminances are printed out (subroutine INTSRF).

Subroutine CALVCP

This routine computes VCP from the previously-accumulated parameters
At this step all that remains to be computed is the factor F, which is computed according to DiLaura[5] by first computing the solid angles subtended by the ceiling (SAC) and the floor (SAF). F is computed for each of the compass directions (four). If an actual viewing direction is not a compass direction, linear interpolation between the bracketing compass directions is used.

Once F has been computed, DGR follows from $DGR = \left[F^{-.44} \sum_i \frac{L_i Q_i}{P_i}\right]^{n^{-.0914}}$

VCP follows from DGR according to DiLaura[5], p. 212.

3.16 OVLY60

Purpose

OVLY60 draws the character contour plots for a rectangular grid of target locations.

Method

If input was in metric units, all dimensions are converted from feet to meters. Then subroutine PLOTS is called once for each plot to be drawn.

3.17 CCMP

Purpose

CCMP maintains the cloudiness database disc file CLOUDS.

Method

Note that the program operates on local file TAPE18. The procedure file CCMPROC saves TAPE18 as CLOUDS when the program terminates.

CLOUDS consists of a number of disc records which is one greater than the number of weather stations currently in the database. The maximum permitted size is 301 records (300 weather stations). Each record contains 41 words. The first word of the first record is the number of stations currently in the database (integer); the remaining 40 words in the first record are ignored.

Record 2 contains the data for weather station 1; record 3 contains the data for weather station 2, etc. The format of each of these records is:

words	contents
1	# overcast days in January
2	# clear days in January
3	# partly cloudy days in January
4	# overcast days in February
5	# clear days in February
6	# partly cloudy days in February
	.
	.
	.
34	# overcast days in December
35	# clear days in December
36	# partly cloudy days in December
37-41	5 words (50 characters) which describe the weather station

An entry is added by writing an additional record at the end of the existing entries. Deleting an entry causes all records beyond the deleted one to be advanced one record toward the beginning of the file, so that no gaps are ever present. This means that deleting weather station J causes all current entries beyond J to be renumbered.

3.18 OBMP

Purpose

OBMP maintains the obstruction database disc file OBSTR.

Method

Note that the program operates on the local file TAPE14. The procedure file OBMPROC saves TAPE14 as permanent file OBSTR when the program terminates.

OBSTR consists of one disc record 2001 words long. The first word is an integer which is the number of obstruction entries presently recorded in the file. Thereafter, each obstruction entry is assigned 10 words in the file:

Obstruction	words
1	2-11
2	12-21
.	
.	
.	
200	1992-2001

Note then that the capacity of the file is therefore 200 obstructions.

Let I, I+1, I+2, ... , I+9 represent the word locations of one obstruction entry in the database. The contents of each entry are then defined as follows:

word	contents
I	units (real number): 1.=English, 2.=metric
I+1	east-west span of obstruction
I+2	north-south span of obstruction
I+3	height of object
I+4	reflectance of west face
I+5	reflectance of north face
I+6	reflectance of east face
I+7	reflectance of south face
I+8	reflectance of bottom
I+9	reflectance of top

An entry is added by simply appending 10 words behind the last current entry. Deleting an entry causes all entries beyond it to be moved 10 words forward, so that no gaps are ever present. This means that deleting entry J causes all current entries beyond J to be renumbered.

3.19 CEL1PP

Purpose

CEL1PP is the preprocessor -- it conducts an interactive question-and-answer session and generates the disc file CEL1DD which can be input to CEL-1.

Method

The coding is straightforward. Each input item is checked for bounds violations or other inconsistencies. If there are any, the last prompt to the user is repeated. It is important to note that the CEL1PP output is written to local file TAPE7 -- it is not saved as a permanent file in the program itself. The procedure file CEL1FE saves local file TAPE7 as permanent file CEL1DD.

3.20 IFACE

Purpose

IFACE conducts a question-and-answer session at the user's terminal; if all is well, a CEL-1 job is submitted for batch processing.

Method

The coding is straightforward. If output is being routed to a remote batch terminal or is being printed at the central site for subsequent mailing, the auxiliary file SITES is read. Each location is allotted 5 lines in SITES; the locations come in SITES in the same order they are printed out during the interactive session.

The first four lines in SITES contain the mailing address if the output is being printed at the central site. The fifth line for each location contains the remote batch terminal login code. If the contents of this line reads NONE, then the user is not given the option of routing his output to a remote batch terminal.

3.21 CELIST

Purpose

CELIST reads a disc file and prints it at the user terminal.

Method

CELIST is intended to print files which contain carriage-control characters in column 1 of each line. Such files are normally generated as if they were destined for a high-speed printer.

CELIST has a "software" top-of-form; i.e., the terminal in question need not be equipped with the form feed feature. Each line of the input file is read into memory. The first character in the line is stripped off and examined. The following actions occur, based on the first character:

1st char	action
1	paper goes to top of form, then the line is printed
0	one line is skipped, then the line is printed
any other character	the line is printed with no skipping

Note that the first character in the input line is never printed. Note also that trailing blanks are never sent to the terminal.

3.22 CELSOL

Purpose

CELSOL determines whether or not the sun is directly visible at target points, interior sensors, and control target area points. The program also determines the area of all light shelves which are exposed to direct sunlight.

Method

1. The necessary binary disc files are read in from disc storage to main memory.
2. For each time instant for which calculations are performed, the solar position is determined (subroutine SOLPOS).
3. A loop is taken thru all fenestration locations. For each fenestration location:
   a) A partition list is constructed of partition surfaces which may potentially interfere with flux through the fenestration.
   b) The solar effect (visible or not visible) at each point is determined (subroutine TPSUN).
4. The solar component (i.e., the transmittance if visible, zero if not visible) at each point is written to disc for each time instant.
5. Assuming 1 fc horizontal illuminance from the sun, the solar flux on each light shelf is determined (subroutine LUMSHF). These results are written into the /COBAS/ common block, which is then written to disc.

3.23 STRIP

Purpose

STRIP is used in the interactive interface procedure (CEL1II) to remove line numbers from the input file CEL1DD. If CEL1DD contains no line numbers, then no alteration is performed. Line numbers must be stripped from CEL1DD, since the CHECK program presumes no line numbers.

Method

The first line is read from local file TAPE4. If the first character in this first line is a decimal digit, the TAPE4 is copied onto TAPE5. minus line numbers. If the first character in the first line is not a digit, then TAPE4 is copied unaltered onto TAPE5.

Whenever line numbers are present in TAPE4 and the first character after the line number is a blank, then the blank is stripped away along with the line number.

Subroutine PARSE

This routine examines the 80-element array A; each element of A is treated as one character. Variables are set as follows:

FIRST is set to point to (1) the first non-blank, non-digit character in A
or
(2) the second character following the line number
whichever comes first.

LAST is set to point to the last non-blank character in A. LAST is set to 1 if A contains only blanks.

Subroutine EMPTY

This routine initializes to blanks the first character in each word of the 80-element array A.

Function IDIGIT

IDIGIT checks the leftmost character in argument word L. If the character is a decimal digit, IDIGIT is set to 1; otherwise, IDIGIT is set to zero.

SECTION IV

Subprogram Descriptions

4.1 ACC1L (also ESIRAT)

Purpose

ACC1L (used in partitioned and/or daylight environments) accumulates the contribution from a given luminaire at a specified gain to illuminance/ luminance at the target points, and the illuminance at interior sensors, and dimming control target area points.

Method

The contribution from the specified luminaire to all points is obtained from the subroutine CONTR1. The contribution at each point is factored by the gain setting, and the resulting value is added to an array containing the previously-accumulated illuminance/liminance.

Variables / Arrays

HFC - vector containing the current accumulated horizontal fc at the target points.
TE - same as HFC, only contains raw fc with body shadow
BL - same as HFC, only contains background luminance
TL - same as HFC, only contains target luminance
SENFC - same as HFC, only contains illuminance at the interior sensors

The following vectors are set by the subroutine CONTR1 to contain the contribution from one luminaire to the metrics being calculated:

FC - horizontal fc at target points
ET - raw fc with body shadow at target points
LB - background luminance
LT - target luminance
SFC - illuminance at sensors
AFC - horizontal illuminance at dimming control target area points
LUM - integer which identifies the luminaire
GAIN - the gain to be applied to the luminaire
CALESI - 0 if ESI is not required
1 if ESI must be calculated

ESIRAT

Purpose

ESIRAT compute the ESI rating from 8 given ESI values.

Method

Let E_i, for i = 1 thru 8, be the 8 given ESI values. Then

$$\text{ESI rating} = e^{(\ln E_1 + \ln E_2 + \ldots + \ln E_8) / 8}$$

$$= e^{(1/8) \ln(E_1 E_2 \ldots E_8)}$$

4.2 ACCAS

Purpose

ACCAS determines the direct contribution of a given luminaire to each partition surface in the room; this contribution is then added to the contribution obtained from all previously-considered luminaires.

Method

Each partition surface is considered in turn. If the luminaire cannot "see" the surface, no further computation is necessary for that surface (e.g., a west-facing surface which lies entirely to the west of the luminaire cannot be illuminated by that luminaire). If the luminaire can "see" teh surface, a list of partitions which could possibly shadow any part of the luminaire is constructed (subroutine APART). Subroutine MAPAR is then called to compute the contribution to each point on the surface.

Variables / Arrays

XL, YL, ZL - (x,y,z) coordinates of the luminaire
XEW, YNS, HLUM - x, y, and z-dimensions, respectively, of the luminaire
LUR - logical unit # of the disc file which contains the point-by-point illuminance on each surface
LUT - logical unit # of the disc file which contains the point-source LUT's
HEAD, X1, X2, Y1, Y2, Z1, Z2, C1, NROWS, NCOLS, DIR - define the linked-list structure which describes the partitions
FC - array used to contain the accumulated contribution at the target points
T - array used to contain the point-source LUT

4.3 ADBAR

Purpose

ADBAR is used in daylighting calculations to append a window's barrier list to the list of building surfaces. When ray tracing through a window is subsequently performed, rays may strike a window barrier rather than another building surface or the sky.

Method

Each of the 4 barriers on the window is defined as a rectangular surface which may then be appended onto the list of building surfaces. The coordinate locations of each barrier surface is obtained from the barrier dimensions and the window location.

Variables / Arrays

CO - (x,y,z) coordinates of the window opening nearest the origin
WWT - width of the window opening
HWT - height of the window opening
BL, BDIS, BH - describe the barriers
IWALL - surface the window lies on (1, 2, 3, or 4)
NB - number of building surfaces
XB,YB, ZB, STYPE, A, B, NORM - describe the building surfaces

4.4 ADDRS

Purpose

ADDRS adds any reflecting surfaces of a sawtooth, skylight, or clerestory structure to the list of building surfaces. A subsequent ray trace through the fenestration can result in one or more rays striking the surfaces of the structure itself rather than the sky or another building surface.

Method

Each of the 5 possible reflecting surfaces is added to the building surface list. In the case of a sawtooth, vertical rectangular surfaces are assigned the height of the structure as a whole. The sloping surface is "raised" to be horizontal and is treated as the top of the rectangular enclosure.

Where a surface is transmitting, it is still added to the list, but with zero dimensions, so that it can never "catch" a ray.

Variables / Arrays

FTRANS - the transmittances of each face of the skylight / sawtooth/ clerestory structure
FDIM - the dimensions of the skylight / sawtooth / clerestory structure
CO - the (x,y,z) coordinates of the fenestration corner nearest the origin
NB - the number of building surfaces
XB, YB, ZB, NORM, A, B - describe the building surfaces
RR - set to 1. if the surface is reflecting and not transmitting
set to 0. if the surface is transmitting

4.5 ADJFL

Purpose

Following the flux transfer analysis which yields the final average luminance on each interior surface, ADJFL is used to construct point-by-point final luminances on the walls and ceiling. When this has been done, the indirect component calculation will account for a non-constant luminance across those surfaces. ADJFL is employed in partitioned and/or daylight environments.

Method

Let $B = (b_{ij})$ be a matrix which specifies the relative initial illuminance at each point on the surface; this initial illuminance may be due to either luminaires or daylight (but not both). Let AF be the average final luminance as determined by the flux transfer analysis. The fin point-by-point luminance at point (i,j) is determined as follows:

$$L_{ij} = C + H_{ij}$$

where

C = average final luminance - average initial luminance

$$H_{ij} = (b_{ij} \times AI) / AB$$

AI = average initial luminance, AB = average of all entries in B

In other words, the final luminance at each point is determined so that:

1) The luminance at each point consists of 2 components:
   a) a term which represents the difference between the average final and initial luminances. This term is the same for each point.
   b) a term which is proportional to the point-by-point initial illuminance.

2) The average of the final point-by-point luminances equals that computed in the flux transfer analysis.

Variables / Arrays

ICEL - index of the ceiling surface in the list structure
AI - vector contains the negative of the initial luminances on each surface
AF - vector contains the average final luminance as computed by the flux transfer analysis
NDIV - the discretization of the major room dimensions
LU - disc logical unit # where the relative initial point-by-point illuminances reside.
IOFS - 1 if routine is called for luminaires
       0 if routine is called for fenestration
B - array which contains the point-by-point initial illuminances

4.6 ADSORT

Purpose

This routine sorts a linked partition list into ascending or descending order, according to a specified field. Such a sorted list is more efficiently searched when it is guaranteed that a "hit" will occur. The routine is therefore used when ray-tracing for the indirect component calculations, for each such ray must eventually strike a room surface if it does not strike any intervening partitions.

Method

P is set to point to the first partition in the list. The remainder of the list is searched, and the last partition T which belongs before P in the sorted list is placed before P. P is then reset to point to T and the process is repeated until no cell belongs before P. P is advanced to the next cell in the list at that time and the entire process is repeated. When P reaches the end of the list, the list has been sorted into the desired order.

Variables / Arrays

HEAD - points to the first cell in the list of partition surfaces to be sorted
LINK - link fields of the partition list
VAL - field on which to sort the list
AD - 1. = sort in ascending order
2. = sort in descending order

4.7 APART

Purpose

APART examines the complete interior partition list and constructs a subset of partitions which could potentially interfere with the contribution from a given luminaire to a given interior target surface. The use of the subset is advantageous since it reduces the number of partition ray traces which must be performed in determining the luminaires contribution to the target surface.

Method

The selection of the subset is accomplished by enumerating the possible combinations of facing directions and excluding those partitions which do not lie between the luminaire and the target surface. For example, suppose the target surface faces west. Then the following partitions may be excluded:

a) those whose easternmost x-coordinate is less than the x-coordinate of the luminaire.

b) those whose westernmost x-coordinate is greater than the x-coordinate of the target surface.

Variables / Arrays

XL, YL, ZL - (x,y,z) coordinates of the luminaire
XP1, XP2 - x-coordinate limits of the target surface
YP1, YP2 - y-coordinate limits of the target surface
ZP1, ZP2 - z-coordinate limits of the target surface
FACE - direction faced by the target surface
HEAD, X1, X2, Y1, Y2, Z1, Z2, LINK, DIR - define the linked-list structure which describes the partitions.
NHEAD - pointer to output linked-list
NLINK - link fields for output linked-list

4.8 BARFL

Purpose

BARFL is used to compute the final luminance (due to daylight only) on each barrier surface surrounding a window. These final luminances are accessed by the prototype FCT to determine the daylight effect within the room.

Method

The initial illuminance from daylight on each barrier surface is determined by projecting 48 rays from the center of each barrier surface and tracing the rays until they strike the sky, a building surface, or the ground. The rays are spaced within the hemisphere in precisely the same way as those projected from each target point; i.e., at elevation angles 15, 37½, 52½, and 75 degrees, and at azimuth angles 15, 45, ... , 315 degrees.

Each zone surrounding a ray projection contributes (1/48) times its luminance to the initial illuminance on the barrier surface. Once the initial illuminances are known, a flux transfer calculation can be performed to determine the final luminance on each barrier surface. Note that the destinations of the 48-ray projections are known and stored on disc prior to entering this routine.

Variables / Arrays

TAU - transmittance of the window
FLBAR - vector of final barrier luminances (output)
IWALL - surface window lies on (1, 2, 3, or 4)
IP - index of the prototype definition for this window
LU - logical unit # of disc file where the 48-ray destinations and the flux transfer matrix reside
BREF - vector of barrier reflectances
NS, STYPE, XB, YB, ZB, A, B, NORM, SREF - describe the list of building surfaces
ISKY - the sky condition (1 = overcast, 2 = clear, 3 = partly cloudy)
SFCSKY - array of fc values (due to the sky) on each building surface for each of the three sky conditions
SFCSUN - array of fc values (due to the sun) on each building surface
ZL - zenith luminance
SUNAZ - solar azimuth (degrees) relative to room north
SUNEL - solar altitude (degrees) relative to the horizon
IPOLL - assumed to be 1. This variable is intended to identify atmospheric conditions in future enhancements to the package.
ROOMAZ - orientation of room north (Degrees) relative to true north
SINEL, COSEL, TANEL - sine, cosine, and tangent, respectively, of the solar altitude
SINAZ, COSAZ - sine and cosine of the solar azimuth
Q - negative of unit vector from the earth to the sun.
LBAR - vector which accumulates initial illuminance at the center of each barrier surface

4.9 BARVAL

Purpose

BARVAL constructs a "mini" room cavity from a window barrier configuration and then computes the flux transfer matrix which is used by BARFL to determine the final luminance on each barrier surface.

Method

The dimensions of the window and barrier components are used to construct a room cavity protruding from the window. The reflectances of the barrier surfaces (and the transmittance of the window glazing) are used to compute the flux transfer matrix. The details of the computation of flux transfer matrix coefficients are given in the discussion of the WRFTM routine.

Details peculiar to the flux transfer matrix for the barriers include:

1. Since the cavity has no "ceiling", a 5 x 5 matrix suffices.
2. Since the barriers may be of different dimensions and hence may not describe a rectangular enclosure, the cavity which is synthesized is the smallest rectangular cavity which contains all the barrier surfaces.
3. The reflectance of a cavity wall is determined by weighting actual barrier surface reflectance by the fraction of the cavity wall which coincides with the barrier surface.
4. The reflectance of the "floor" is one minus the transmittance of the window.

Variables / Arrays

REFWT - reflectance of window glazing
HWT - height of the window opening
WWT - width of the window opening
BDIS, BH, BL, BREF - describe the barriers at this window
FTM - a 5 x 5 array of coefficients for the flux transfer calculations

4.10 BASET (also subroutine VFBLA)

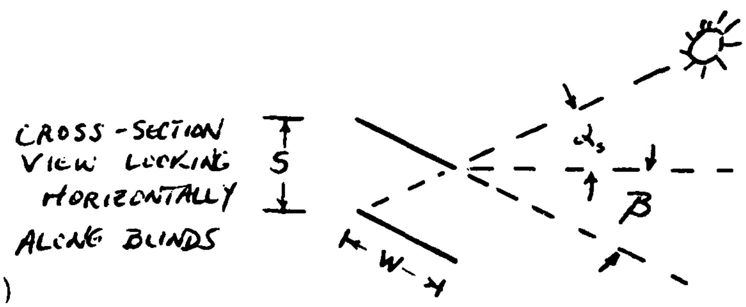
Purpose

This routine computes the threshold blinds angle setting needed to completely shut the sun out of the room. This figure is used in turn to compute the blinds angle setting when the profile mode user has requested that the program do so.

Method

If the sun is not visible to the window containing the blinds, then the blinds setting is full open (i.e., vanes normal to the window surface). Otherwise:

a) Horizontal blinds -



α_s = solar profile angle

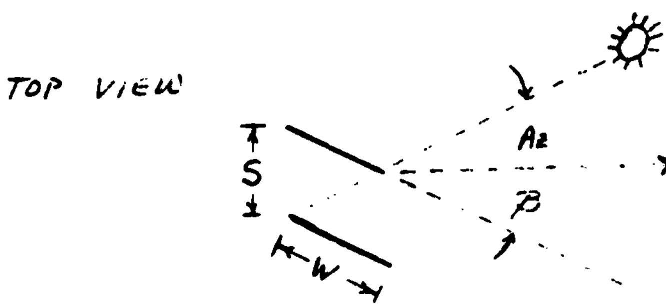
$$= \tan^{-1}\left[\frac{\tan EL}{\cos Az}\right]$$

EL = solar altitude

Az = solar azimuth, relative to normal to the window

$$\beta = \text{blinds angle setting} = \max\left\{0,\ \sin^{-1}\left[\frac{S\cos^2\alpha_s - \sqrt{1-\cos^2\alpha_s}\ \sqrt{W^2 - S^2\cos^2\alpha_s}}{W}\right.\right.$$

b) Vertical blinds -



$$\text{Blinds angle setting} = AN - SG(Az)\,\sin^{-1}\left[\frac{S\cos Az - \sqrt{1-\cos^2 Az}\ \sqrt{W^2 - S^2\cos^2 Az}}{W}\right]$$

where AN = Angle of $\vec{n}$ SG = +1 if Az > 0, -1 if Az < 0

VFBLA

Purpose

This routine returns vertical fc on a window surface for given daylight conditions; the threshold blinds angle setting required to shut the sun out of the room is also returned.

Method

From the day of the year and the time of day, the solar position is determined. From the solar position, BASET may be called to compute the threshold blinds angle setting. The vertical fc calculation employs the subroutines HFCSUN, HFCSKY, and FCBS. The vertical calculation is:

$$E = E_{vsky} + E_{vsun} + \rho_g (E_{hsun} + E_{hsky})$$

where

E_{vsky} = vertical fc due to clear sky

E_{vsun} = vertical fc due to sun

E_{hsky} = horizontal fc due to clear sky

E_{hsun} = horizontal fc due to sun

ρ_g = reflectance of the ground

Variables / Arrays

IMONTH - month of the year (1 thru 12)
JD - Julian day
TIME - time of day
IWALL - wall the window lies on (1, 2, 3, or 4)
ITYPE - (1 = horizontal blinds, 2 = vertical blinds)
S - spacing between blinds vanes
W - width of one blinds vane
DSTMAP - daylight savings time map
LAT - latitude
ROOMAZ - room azimuth, relative to true north
RHOG - reflectance of the ground
ANGLE - threshold blinds angle setting (output)
VFC - vertical fc outside the window (output)
NSURF - the number of building surfaces
NORM - unit outward normal vector from each building surface

Variables / Arrays

ITYPE - (1 = horizontal blinds, 2 = vertical blinds)
IWALL - wall the window lies on (1, 2, 3, or 4)
ROOMAZ - azimuth of room north, relative to true north
SUNAZ - azimuth of sun, relative to room north
S - spacing between blinds vanes
W - width of one blinds vane

4.11 BGRAH

Purpose

BGRAH plots the hourly energy consumption in histogram form on the printed output.

Method

First, the mean, maximum, minimum, and standard deviation are determined for each hour during the day. Let N_o, N_c, N_p represent the expected energy consumption on an overcast, clear, and partly day, respectively. Then if OD, CD, PD respectively represent the number of overcast, clear, and partly cloudy days in the month, then the mean energy consumption is:

$$(N_o OD + N_c CD + N_p PD) / (OD + CD + PD)$$

The mean, maximum, and minimum and standard deviation are also expressed in percent of that possible with all luminaires operating.

Each hour is allotted 101 character positions across the page; the leftmost character position corresponds to zero percent, while the rightmost character position corresponds to 100%. The zero to minimum consumption range is plotted as a string of '-' characters; the minimum to maximum range is plotted as a string of '+' characters. An 'O' character locates the mean.

Variables/ Arrays

ENER - 24 x 3 array giving the power consumption for each hour for each of the three cloud conditions. The units are kilowatts, but also may be thought of a kilowatt-hours since they are for a time period of one hour.

PROB - 24 x 3 array giving the power consumption for each hour for each of the 3 cloud conditions. In this implementation, these probabilities are the same for each hour.

ID - 5 lines of label information (not used in this implementation)

MONTH - the month of the year (not used)
LU - logical unit # of printed output
RDMAX - maximum possible power consumption, based on all luminaires operating
IHR1, IHR2 - first and last hours room is occupied
EMEAN, EMAX, EMIN, ESDEV - statistics for consumption, in kW (or kW-hours)
PMEAN, PMAX, PMIN, PDEV - statistics for consumption, in % of possible
IDMEAN, IDMAX, IDMIN - character positions of each statistical value

4.12 BILFSL

Purpose

This routine constructs the arrays which describe the room's fenestration.

Method

The routine loops thru the arrays which are read directly from the input stream and constructs other arrays and variables which are more easily processed in subsequent calculations. Among these latter are:
NW - the total number of fenestration elements (windows + skylights, etc.)

WC1 - 3 x NW array giving the (x,y,z) coordinates of the corner of each fenestration element which is nearest the origin.

WC2 - 3 x NW array giving the (x,y,z) coordinates of the corner of each fenestration element which is farthest from the origin

WPRO - vector which points to the prototype definition corresponding to each fenestration element

NPRO - the number of fenestration prototype definitions. Two or more fenestration elements share one prototype definition if they are of the same type (e.g., both are windows) and they lie on the same room surface.

PROSRF - vector giving the surface # of each prototype definition.

PROTYP - vector gives the fenestration type of each prototype definition: 1 = window, 2 = clerestory, 3 = sawtooth, 4 = skylight

Other Variables / Arrays

NFTYPE - the number of fenestration sub-blocks which have been read from the input stream
FTYPE - the type of fenestration associated with each fenestration sub-block in the input stream. Coded the same as PROTYP, above
NFLOC - the # of locations associated with each input sub-block.
FDIM - the fenestration dimensions given in each sub-block.
FSTRT - the (x,y,z) coordinates nearest the origin of each fenestration element in the sub-block.
FTRANS - transmittances of each fenestration sub-block.
FSURF - the surfaces which the fenestration elements defined in each sub-block lie on.

4.13 BILPAR (also INCEL, GETCEL, RETCEL)

Purpose

BILPAR reads the obstruction specifications from the user's input file and constructs a "partition list", where each list entry is a planar surface which may be one face of an obstruction, an insert, or a room surface.

Method

First, the linked list structure which is to describe the partitions is initialized (subroutine INCEL). The obstruction database is then read into memory from disc; the dimensions of the database entries are converted to feet, if necessary.

The location of each user-defined obstruction is read in; the subroutine OBSURF is called for each one. OBSURF breaks the obstruction into its component surfaces and adds each surface to the partition list. Each partition surface is discretized to provide zone sizes the same as (within roundoff) those specified for the room surface which is parallel to the partition surface. The resulting discretization is recorded as the number of rows and columns of discrete points on the surface.

Finally, the room surfaces and inserts are added to the list. Inserts are assigned coordinates in such a way that they lie incrementally closer to the center of the room than the wall they lie on. In this way an insert completely blocks light rays destined for the portion of the wall it covers.

Variables / Arrays

LU5 - logical unit # for the user input stream
LUIOS - logical unit # for the disc file where accumulated initial illuminances on partition surfaces are stored
LUPAR - logical unit # for the disc file where the linked-list structure describing the partitions is written
ISW - 0 if no obstructions are present within the room
1 if there are obstructions present within the room
RHOIN - vector gives the reflectance of each insert/fenestration opening
LIMIN - array gives the coordinate limits of each insert/fenestration opening.

NOTE: For purposes of constructing the partition list, fenestration openings are treated as inserts.

INCEL

Purpose

INCEL initializes the pool of available cells (for partition list structure). The routine is called once before construction of the partition list and need never be called again.

Method

The calling program supplies three parameters:
IAV - will become the pointer to the list of available cells
LINK - the link field which joins cells to each other
N - the number of cells which should form the available pool

Each cell except the N^{th} is linked to the next sequential member of the array -- i.e., LINK(I) = I+1. The link field of the N^{th} cell is set to zero, signifying the end of the list. IAV is set to point to cell 1.

GETCEL

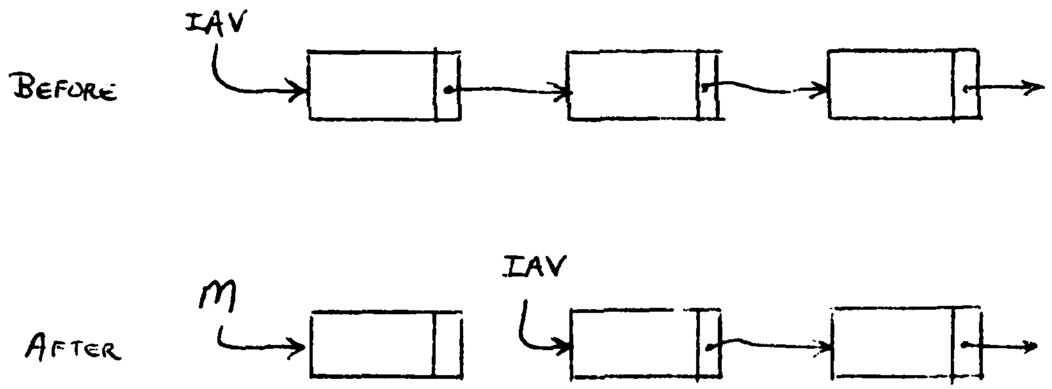
Purpose

GETCEL fetches the next available cell (for partition list structure from the pool of available cells.

Method

The calling program supplies an integer variable to serve as a pointer to the next available cell (M). GETCEL sets M to point to the first cell in the available pool. The pointer to the first available cell (IAV) is then set to point to what was formerly the second available cell

The sketch below shows the status of the available pool before and after a call to GETCEL.



RETCEL

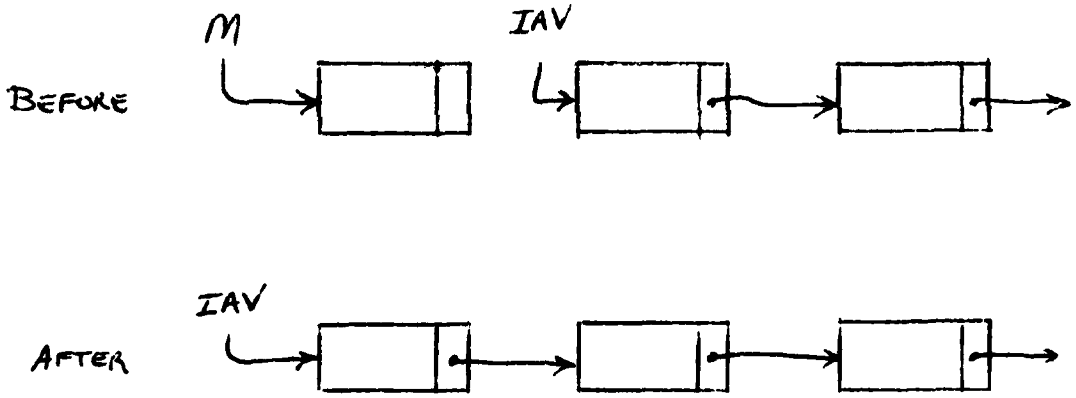
Purpose

Occasionally, a cell (for partition list structure) will be released by GETCEL and it will subsequently develop that the cell is not needed. In these cases RETCEL is used to return the cell to the pool of available cells.

Method

The integer variable M points to the cell which is being returned by the calling program. This cell becomes the first cell in the updated pool of available cells; the cell which was formerly first on the list becomes the second cell on the list.

The sketch below shows the status of the available pool before and after a call to RETCEL.



4.14 BILTAS

Purpose

BILTAS is called in partitioned and/or daylight environments to compute target points and viewing directions whenever unknown task locations (i.e., rectangular grid) are not being used. For TASK=KNOWN, the location and viewing direction will already be completely specified; however, for TASK=RATING the BILTAS routine explodes the given task location and primary viewing direction into the eight target points and viewing directions necessary for ESI Rating computations.

Method

A loop counter is set to 1 (TASK=KNOWN) or 8 (TASK=RATING). Each trip through the loop a target point with its unique viewing direction is generated and added to a linear list of such points. The viewing direction is determined either from the IES specification for ESI rating viewing directions, or from the Navy option for these viewing directions (refer to the User's Guide for details on the difference).

Variables / Arrays

XP, YP - (x,y) coordinates of the input task location
VIEW - the primary viewing direction for the input task location
IOP - (1=use IES option for ESI rating view directions;
2=use Navy option for ESI rating view direction)

XTP, YTP - (x,y) coordinates of target points to be generated by this routine

VIEWD - viewing directions at target points; calculated by this routine

LINKT - link field for target points belonging to one task location

LINKV - link field for target points having the same viewing direction

HEADT - points to first cell on list of target points organized by task location
HEADV - points to first cell on list of target points organized by view direction

AVAIL - points to first cell on pool of available cells

IKNOWN - (1=ESI ratings, 3=known task locations, no ESI rating)

CX,CY,SX,SY - arrays define the offsets from the task location used to compute the target points needed for ESI rating computations

INC - array specifies the increments of $22\frac{1}{2}^{\circ}$ used to compute the viewing directions at the 8 target points for ESI rating calculations

4.15 BLILUM

Purpose

In a daylighting environment where blinds are used, BLILUM is used to calculate the final luminance evident inside the room. The luminance so determined is used to complete the prototype fenestration candela table.

Method

Calculations proceed in the following order (horizontal blinds):

1. The direct sunlight effect on the lower of two adjacent vanes is determined.
2. The sky effect on the lower blind is determined.
3. The ground contribution on both the lower blind and on the lower portion of the upper blind is determined.
4. Using the contributions from 1-3 (this yields the initial illumination on the facing surfaces of any two adjacent vanes), a "mini"-flux transfer analysis is performed to compute the final luminance on the lower surface of the upper blind -- this is the brightness evident within the room.

For the vertical blinds case, the calculations are similar, except that the sky and ground each contribute to both vane surfaces.

Variables/ Arrays

BTYPE - blinds type (1=horizontal, 2=vertical)
IS - surface number of the window containing the blinds (1,2,3, or 4)
BANG - the blinds angle setting
W - width of one blinds vane
S - spacing between adjacent vanes
T - thickness of one vane
LG - luminance of the ground
FCSUN - vertical fc on surface IS due to the sun
RHOB - reflectance of the blinds
IT - the prototype definition identifier for the window in question
LZ - zenith luminance
ISKY - sky condition (1=overcast, 2=clear, 3=partly cloudy)

4.16 BLSKYV

Purpose

This routine computes the fraction of a window piece through which the sky is visible, based on the presence of venetian blinds at a given angle. This fraction is used in daylight calculations to split the window's contribution between a diffuse FCT and a clear sky FCT. Note that this fraction varies with target point location relative to the window piece.

Method

A ray is projected from the target point to the center of the window piece. The elevation angle of this projected ray is used to compute what portion of the spacing between two adjacent vanes is visible through the window to the sky, as in the following formula:

$$\beta = \begin{cases} \tan^{-1}\left[\dfrac{p_x - b_x}{b_z - p_z}\right] & \text{window in west wall} \\ \tan^{-1}\left[\dfrac{b_y - p_y}{b_z - p_z}\right] & \text{north} \\ \tan^{-1}\left[\dfrac{b_x - p_x}{b_z - p_z}\right] & \text{east} \\ \tan^{-1}\left[\dfrac{p_y - b_y}{b_z - p_z}\right] & \text{south} \end{cases}$$

$$\omega = \beta - \text{blinds setting}$$

$$BLSKYV = 1 - \min\left\{1, \frac{1}{S}\left(W|\cos\omega| + T|\sin\omega|\right)\right\}$$

Vertical blinds are analogous.

p = P below
b = B below
T = thickness of vane.
W, S defined below.

Variables / Arrays

B - (x,y,z) coordinates of the center of the window piece
P - (x,y,z) coordinates of the target point
ANG - angle setting of the blinds
W - width of one vane
S - spacing between adjacent vanes
TYPE - (1=horizontal blinds, 2=vertical blinds)
IS - surface # (1, 2, 3, or 4) of window

4.17 BPART

Purpose

BPART is almost identical to APART; both routines construct a list of partitions prior to computing the illuminance on a target surface (room surface, fenestration rectangle, insert, or obstruction face). The difference is that the emitting source for BPART is a fenestration rectangle. For APART the emitting source is a luminaire. For details on the algorithm refer to the documentation for APART.

4.18 BRDF

Purpose

For purposes of ESI calculations, body shadow tables and BRDF factors are usually given in spherical coordinates. The BRDF routine converts such tables into tables of shadow and BRDF factors which have tabulated values at the asymmetric LUT ordinates. Thus, shadow and BRDF factors can be accessed without expensive trigonometric manipulations; instead, they can be accessed in two linear dimensions via an LUT.

Method

First, the routine loops through the 21 x 21 points which constitute the LUT. From each point, the azimuth and elevation angles to the table origin are computed, based on a zero-degree viewing direction. These azimuth and elevation angles are stored in the arrays PHI and THETA, respectively.

Next, the routine loops on the 16 possible viewing directions (0^{o}, $22\frac{1}{2}^{o}$, ... ,$337\frac{1}{2}^{o}$). For each of these viewing directions, all 441 LUT points are enumerated once more. For each point, the elevation angle is already available from the THETA array; the azimuth angle is obtained simply by subtracting the viewing direction from the azimuth stored in the PHI array.

Variables / Arrays

T - 19 x 37 x 3 array of shadow and BRDF factors, vs. spherical coordinates. T(*,*,1) is body shadow; T(*,*,2) is background luminance BRDF; T(*,*,3) is target luminance BRDF.

R - 21 x 21 x 3 array of output values computed at the asymmetric LUT points; the order of the 21 x 21 planes is the same as in the T array.

LU - logical unit # of the disc file where the computed tables R are to be written.

PHI,THETA - 21 x 21 arrays discussed in "Method".

4.19 CANDMT

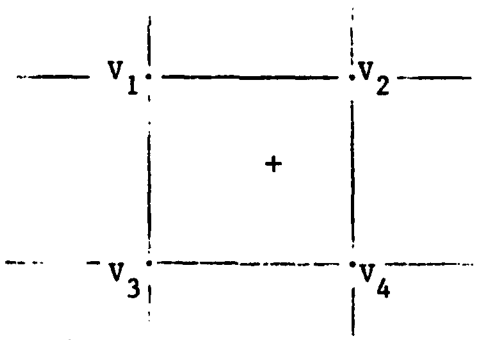
Purpose

CANDMT computes the multipliers which apply to each corner of a cell of a two-dimensional interpolation table. The 4 multipliers apply to a given target point, and to a given point source, which will be a discrete piece of wall or ceiling fenestration. For a given target point, the aggregate of all calls to CANDMT (one call per discrete fenestration piece) result in the complete fenestration candela table (FCT).

Method

The target point is treated as the origin of an asymmetric LUT. The displacement of the window point in the LUT will isolate a particular cell in the LUT. The LUT values ultimately will be luminance evident when looking out the window. Since these luminance values change from sky condition to sky condition, but none of the other pertinent factors (geometrical relationships between target point and fenestration) do change, the idea of CANDMT is to complete the calculation as much as possible when lacking only the LUT values.

To this end, let + denote the displacement of the window point in the LUT:



If the luminance values V_1, V_2, V_3, V_4 were known, it would be possible to compute the illuminance at the target point due to the window piece in question. Since the V_i are not known, however, our goal is to compute as much of the final result now (once), so that the future computation of the final result (when the V_i are known) will be as efficient as possible. Such efficiency is important, since we will need many results for many different values of the V_i.

The discussion of the FCT (section 2.2) gives the formulae used in CANDMT to compute the multiplier values for each corner point in the LUT cell.

4.20 CHROOM

Purpose

CHROOM is used to determine the parameters used to scale a character plot of the room. The routine returns the scale (feet or meters per inch) to use, and the number of columns and rows to use in constructing the plot.

Method

A fixed list of possible scales is searched to determine the largest which will fit the entire room on an 8" x 8" area. If the room is so large that none of the fixed possible scales will accomodate the room, then the scale is set to the largest integer multiple of 10 (6.517 for metric) which will contain the room.

Variables / Arrays

RSCAL - 12 x 2 array of scales to try. First column is for English units, second column is for metric units. Scales are feet per inch for English, meters per inch for metric.

4.21 CLUT

Purpose

CLUT computes point source LUTs from a luminaire to all six surrounding surfaces. These LUTs may be used to compute the contribution from a discrete piece of luminaire to any interior surface.

Method

The point source is placed at the center of an imaginary cube 2' on a side. POint source LUTs for illumination are then computed on each of the six inside surfaces of the cube. These six LUTs are written to the disc file whose logical unit # is given by the parameter LU12.

The inverse square law is used to calculate the illuminance at each point in the LUTs. CVndela values for the inverse square law application are obtained from the 2-dimensional array CD by using the index tables TNDX and PNDX (see the discussion of subroutine PHOTO). The orientation of the luminaire is accounted for by constructing a 3 x 3 rotation matrix A which transforms (x,y,z) coordinates given in system S_1 to the equivalent coordinates given in S_2. EAch system has its origin at the center of the cube; the S_1 axes are:

+x = east   +y = north   +z = up

The S_2 axes reflect the actual positioning of the luminaire:

+x = 270° photometric plane   +y = 0° photometric plane

+z = photometric nadir

If (u,v,w) are the coordinates of a point in system S_1, then the corresponding coordinates (x,y,z) in S_2 are given by:

$$x = A_{11}u + A_{12}v + A_{13}w$$

$$y = A_{21}u + A_{22}v + A_{23}w$$

$$z = A_{31}u + A_{32}v + A_{33}w$$

The virtue of having (x,y,z) is that the azimuth Ø and declination angles θ which locate the desired candela values are then easily computed as

$$\emptyset = \text{ATAN2}(-x,y)$$

$$\theta = \text{ATAN2}(\ (x^2 + y^2)^{\frac{1}{2}},\ z)$$

4.22 CMPRES

Purpose

CMPRES is used in a daylighting environment to minimize the disc accesses which must be performed to determine the luminaires' contribution. As many "quasi-luminaire" contributions are packed into one disc record as possible; this number will depend on the number of target points, interior sensors, area points, and whether or not ESI is required.

Method

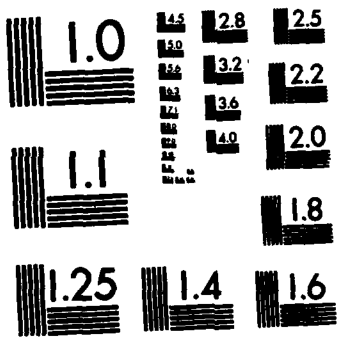
Calculations are made to determine how many "quasi-luminaire" contributions will fit on one disc record; these contributions are then packed into the vector R.

Variables / Arrays

LU - logical unit # of disc file where contributions from each "quasi-luminaire" are stored
LUCMPR - logical unit # of disc file to receive output from this routine
NLUMS - # luminaires
NTP - # target points
NSEN - # interior sensor locations
NAPTS - # area points
DOESI - (0=don't compute ESI; 1=do compute ESI)
NWPL - # words per luminaire on an output disc record
NWPR - # words per output disc record
NLPR - # luminaire contributions per output disc record
R - vector which contains one output disc record (= contribution from NLPR luminaires)
MAXLEN - maximum allowable word length per output record

AD-A124 215 CEL-1 LIGHTING COMPUTER PROGRAM - PROGRAMMER'S GUIDE 2/3
(U) F AND K GROUP NEW YORK W E BRACKETT JAN 83
NCEL-CR-83.009 N68305-80-C-0012

UNCLASSIFIED F/G 9/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

4.23 CMTFES

Purpose

CMTFES constructs the FCT from one fenestration type (all elements) onto an interior surface.

Method

A loop is made through the list of all fenestration elements. Each one of the given type is isolated and its contribution is accumulated. If shades or drapes are present, their area must be computed and accounted for. If blinds are present, the routine BLSKYV is invoked to determine what fraction of blinds and what fraction of sky are visible.

When the FCT has been computed, the vectors LIMCOL and LIMROW are set so as to identify the smallest rectangular subset of the FCT which contains all of the FCT's nonzero entries.

4.24 CONMAP

Purpose

This routine constructs the parameters which are used to manipulate fenestration contribution. The resulting arrays are called the fenestration map entries.

Method

A fenestration map entry isolates a subset of the fenestration elements which share common properties:

1. Same type (all windows, all skylights, etc.)
2. Same glazing type
3. Same room surface
4. Same arrangment of blinds, shades, drapes, light shelf

A window which contains blinds, shades, or drapes spawns two map entries: one for clear glazing and one for diffuse.

In operation, the routine loops thru each fenestration element (window, etc.) and checks to see if a map entry which describes the entry already exists. If not, such an entry is added to the list. ULtimately, one FCT must be computed for each map ent-y.

Variables / Arrays

NMAP - # map entries (output)
MAPFT - pointer to the prototype entry (PROTYP) which in turn gives the fenestration type (1=window, 2=clerestory, 3=sawtooth, 4=skylight)
MAPSU - room surface number associated with each map entry
MAPGL - identifies the glazing of each entry:
- 1 - clear, no blinds
- 2 - diffuse, may be drapes, shades, or diffuse glass
- 3 - clear, visible thru blinds
- 4 - diffuse, results from blinds

WPRO - identifies the fenestration prototype entry associated with each fenestration source element
PROTYP - gives the fenestration type of each prototype entry (1=window, 2=clerestory, 3=sawtooth, 4=skylight)
PROSRF - gives the surface number for each prototype entry
FSHR - gives the glazing for each fenestration type (1=clear, 2=diffuse)
SHADET - transmittance for shades
SHADED - pull-down depth for shades
DRAPET - transmittance for drapes
DRAPED - close distance for drapes
FDIM - dimensions of each fenestration type
NW - # of fenestration source elements
BLTYPE - type of blinds (1=horizontal, 2=vertical)

4.25 CPARTL

Purpose

CPARTL builds a list of partitions which can interfere with the contribution of a specified point source into a specified quadrant.

Method

The parameter IQ specifies which quadrant the list is to be built for:

```
      !
  2   !   1
      !
  ----+----
      !
  3   !   4
      !
```

The list of all partitions is searched. Those are added to the output list which protrude into the quadrant anywhere within the outer limits of the target points. The output list cells are joined together by the link field NLINK; NHEAD points to the first cell on the list.

4.26 CONTR1

Purpose

CONTR1 fetches from disc the contribution from a given "quasi-luminaire" (see discussion of subroutine CMPRES) to target points, interior sensors, and area points.

Method

The record containing the desired contribution is read into memory (unless it already resides in memory). The record is disected so that its contribution can be allocated to the calling program's arrays.

Variables / Arrays

ILUM - # of the "quasi-luminaire" whose contribution is desired
FC - illuminance at the target points
ET - raw fc with body shadow at the target points
LB - background luminance at the target points
LT - target luminance at the target points
SFC - illuminance at the interior sensors
AFC - illuminance at the area points

4.27 CPRO48 (also RAY6)

Purpose

CPRO48 is used in computing initial illuminance on the top surface of clerestory, sawtooth, or skylight fenestration. This value is ultimately used as input to a flux transfer analysis which determines the final luminance on the interior surfaces of fenestration of these types.

Method

In a fashion similar to that used for computing the indirect component on target points in a partitioned environment, 48 rays are projected from the center of each surface and are traced to their destination. Each ray may strike another surface of the fenestration structure, the sky, any building surface, or the ground. The (imaginary) bottom of the fenestration structure is treated as a possible ray destination in this scheme.

Arrays which completely characterize the ray destinations are written. The flux transfer matrix is also computed.

Variables / Arrays

HS - gives the surface # of the destination of each projected ray:
- 0 - ray hit sky
- n (positive integer) - ray hit building surface n
- -n (negative integer) - ray hit surface n on the fenestration structure

HX - x coordinate of ray destination, if a building surface. azimuth angle, if sky

HY - y coordinate of ray destination, if a building surface. elevation angle, if sky

HZ - z coordinate of ray destination, if a building surface; ignored if sky

HCOS - cosine between projected ray and normal to building surface

4.28 CTIMES

Purpose

CTIMES computes the 15 time instants during the year when daylight effects are to be computed. The resulting computations are used to determine the energy profile.

Method

Three days are selected: Dec. 22, March 21, and June 21. On each of these days 5 times are chosen: sunrise, sunset, plus three additional times which partition the solar day into 4 equal periods.

4.29 CTSURF

Purpose

The routine computes a candela multiplier table from a fenestration point to an interior surface. The table, when multiplied by exterior luminance at the LUT points, yields the average illumination on the surface due to the fenestration piece. The sum of all such multiplier tables (accounting for all fenestration pieces) constitutes the FCT for the target surface from the fenestration type in question.

Method

A list of partitions is constructed which might possible interfere (subroutine APART). CANDMT is then called once for each point on the target surface. The results of CANDMT are divided by the number of points on the surface and are added together to produce the output candela multiplier table.

4.30 CVMET

Purpose

CVMET converts illuminance / luminance values from metric to English units.

Method

The following conversion formulae apply:

Illuminance: 1 fc = 10.764 lux

Luminance: 1 fL = 3.426 candelas / $meter^2$

Variables / Arrays

FC - horizontal illuminance at the target points
ET - raw illuminance with body shadow at the target pts
LB - background luminance at the target pts
LT - target luminance at the target points
ESI - ESI at the target pts
ESIR - ESI Ratings at the task locations
APFC - illuminance at area points
SFC - illuminance at interior sensors
ESFC - illuminance at exterior sensors

4.31 DAYEF

Purpose

For given sky conditions, DAYEF computes the daylight effect on target points, area points, interior sensors, and exterior sensors.

Method

The following steps are performed:

1. Zenith luminance and illuminance on all building surfaces are computed. Note that for the overcast sky the zenith luminance is arbritrarily set to 1000.

2. Subroutine FILPRT is used to complete each prototype FCT; i.e., the actual exterior luminance at the 441 LUT points is determined.

3. The average initial illuminance at each interior surface is computed.

4. Flux transfer analysis yields the final average luminance on each room surface.

5. The final zone-by-zone luminance is determined on the walls and ceiling.

6. The final interior surface luminances and the destinations of the 48-ray projections are used to determine the indirect daylight component at the target points, area points, and sensors.

7. The FCTs for direct contribution are read in, and the direct daylight component is computed at the target points, area points, and sensors.

4.32 DIGITZ

Purpose

For computing energy profiles, this routine is used to adjust an interpolated energy consumption value which has been computed without regard to the dimming method used. For example, with high-low-off switching, it is not possible to operate a luminaire at .35 gain. DIGITZ "digitizes" energy consumption so that the gain resolution which prevails is consistent with the total possible energy consumption.

Method

The input energy consumption (E) is rounded to the nearest integer multiple of the discrete increment (DEL). This value is then forced to be no less than the minimum energy (EMIN) which must be consumed. The value is also not allowed to exceed the consumption when all luminaires are operated at full gain.

4.33 D048

Purpose

D048 projects 48 rays from each target point, interior sensor, and area point. The rays are traced until they strike an interior surface; the destinations of all rays are recorded on disc for later retrieval.

Method

The 48 rays are projected from each point at all 48 combinations of the 4 elevation angles 15, 37½, 52½, and 75 degrees and the 12 azimuth angles 15, 45, ... , 315 degrees. The elevation angles are relative to the plane containing the point in question; the azimuth angles are relative to room north. The actual ray tracing is performed by the subroutine PROJ48.

Variables / Arrays

HEADT - points to the first cell in the list of target points organized by task location
XTP,YTP - x and y coordinates of the target locations
LU48 - disc file logical unit # where ray-trace destinations are to be written
SNID - vector contains the direction faced by the interior sensors
SNIC - 3 x 10 array containing the (x,y,z) coordinates of the interior sensors
APX,APY - x and y coordinates of the area points
TORGAP - (x,y,z) coordinates of the area point nearest the origin

4.34 DR48

Purpose

DR48 projects 48 rays from the center of each surface comprising a barrier cavity. This is done so that a flux transfer analysis can be performed on the cavity to determine the final luminance on each barrier surface. The destinations of the ray projections are written to disc for later retrieval.

Method

First, the 4 barrier surfaces are added to the building surfaces list. (This is so that the barrier surfaces may be treated just like building surfaces by the ray-tracing subroutine RASTRK)

48 rays are projected from each surface at elevation angles 15, 37½, 52½, and 75 degrees and azimuth angles 15, 45, ... , 315 degrees. The elevation angles are relative to the barrier surface in question; the azimuth angles are relative to an outward normal from the window surface. The rays are traced till they strike either of:

1) the sky (i.e., no surfaces)

2) another barrier surface

3) a building surface

4) the ground or a ground insert

4.35 EQTIME

Purpose

The EQTIME routine provides the correction factor needed to account for the effect of the eccentricity of the earth's orbit on the relationship between local time and solar time.

Method

Let L be the local time before the eccentricity correction is applied. Then the actual local time T is:

$$T = L + .1293 \sin(2\pi(JD-2.265)/365) + .1569 \sin(4\pi(JD+10.60)/365)$$

where JD is the Julian Day (1=Jan. 1, 2=Jan. 2, ... , 365=Dec. 31)

4.36 ES48

Purpose

ES48 performs the preliminary calculations necessary to compute the reflected illuminance component at the exterior sensor locations. 48 rays are projected from each exterior sensor location and their destinations saved for later retrieval.

Method

First, it is determined whether the exterior sensor lies within a window barrier cavity. If so, the barriers are added to the building surface list. 48 rays are then projected from the sensor location; each ray is traced until it strikes one of:

1) a barrier surface
2) the sky
3) the ground
4) a building surface

The destinations are recorded. At a later time, when the luminance at all the ray destinations becomes known, the reflected component at the sensor location is

$$\frac{1}{48} \sum_{i=1}^{48} L_i$$

, where the summation is over the 48 luminances at each ray destination

Variables / Arrays

SLOC - (x,y,z) coordinates of the exterior sensor
SDIR - direction the sensor faces
IP - points to the fenestration prototype which describes the window nearest to this sensor. IP=0 if there is no fenestration on the wall where the sensor is located.
WC1,WC2 - the lower and upper (x,y,z) coordinates of the window in question.
PROTYP,PROSRF - the fenestration type and surface # for prototype IP.
BDIS,BL,BH - barrier dimensions
WWT,HWT - width, height of the window
HS - identifies the destination of each ray:
  0 - sky
  positive integer - building surface
  negative integer - barrier surface
HX,HY,HZ - (x,y,z) coordinates of ray destination on a building or barrier surface. If sky is destination, HX = azimuth angle, HY = elevation angle, HZ is ignored.
HCOS - not used

4.37 FCBS

Purpose

FCBS computes illumination due to the sun and sky on the building surfaces.

Method

Horizontal illuminance due to the sun (HSUN) and to the unobstructed sky (HSKY) are input to the routine. In this routine the shadowing effect of buildings on each other is ignored.

For horizontal surfaces facing the sky, the illuminance values are simply HSUN and HSKY. Vertical illuminance from the sky is computed by the subroutine VSKY. Vertical illuminance from the sun is given by:

$$V_{sun} = \frac{(n_x \sin(Az) \cos(El) + n_y \cos(Az) \cos(El))\ HSUN}{\cos(El) \tan(El) (n_x^2 + n_y^2)^{\frac{1}{2}}}$$

where n_x = x-component of outward normal from the surface

n_y = y-component of outward normal from the surface

Az = solar azimuth relative to room north

El = solar elevation

4.38 FCESEN

Purpose

FCESEN calculates the illuminance on the exterior sensors.

Method

The 48-ray destinations from each sensor are retrieved from disc (these destinations will have previously been calculated in the subroutine ES48). The flux transfer analysis which yields the final luminance on the surrounding window barrier surfaces is performed. (The illumination on building surfaces is passed as an argument to the routine.)

The destination of each ray is examined and the luminance at that point is multiplied by 1/48. The sum of the 48-ray contributions is the illuminance at the sensor. Note that if a ray strikes a building surface, a ray-trace is performed to see if the sun illuminates the surface at that point.

Variables / Arrays

- LUS48 - logical unit # of the disc file which contains the destinations of the 48 rays projected from the sensors.
- LUB48 - logical unit # of the disc file which contains the destinations of the 48 rays projected from each barrier surface, plus the flux transfer matrix for the barrier cavity.
- ESFC - output vector of illuminance on the exterior sensors.
- ISKY - sky condition (1=overcast, 2=clear, 3=partly cloudy)
- SFCSKY - 3 x 30 array gives the illuminance on the exterior surfaces due to the unobstructed sky.
- SFCSUN - 30-element vector which gives the illuminance on the exterior surfaces due to the sun.
- LZ - zenith luminance
- SNEW - vector which identifies the window which is nearest to each sensor. If no window is on the same wall as the sensor, the corresponding entry in SNEW is zero.

4.39 FCTDSK

Purpose

This routine computes all FCTs for interior surfaces, area points, and interior sensors.

Method

The outer loop is on fenestration map entries (see discussion of the CONMAP subroutine). For each map entry, one FCT will be written for each interior surface, each area point, and each interior sensor. I.e., the total number of FCTs generated in FCTDSK will be

(# map entries) x (# interior surfaces + # area points + # interior sensors)

Variables / Arrays

LUFCTS - the logical unit # of the disc file where the computed FCTs are to be written

ORG - the (x,y,z) coordinates of the point on each interior surface which is nearest the origin

FCT - the 21 x 21 word array used to store each FCT

LSCOL - a 2-element vector giving the leftmost and rightmost columns, respectively, of the smallest rectangular subset of the FCT which contains all of the FCT's nonzero entries.

LSROW - a 2-element vector giving the lower and upper rows, respectively, of the smallest rectangular subset of the FCT which contains all of the FCT's nonzero entries.

Note that each disc record contains 445 words: the 441-word FCT array, followed immediately by LSROW and LSCOL.

4.40 FCTTP

Purpose

FCTTP computes a 4-entry multiplier table for a given target point from a point fenestration source. The sum of all such multiplier tables for each fenestration piece making up the map entry becomes the FCT from that map entry to the target point.

Method

CANDMT returns the 4 multipliers giving horizontal fc at the target points. If ESI is being computed, factors for raw fc w/body shadow, background luminance, and task luminance are obtained via interpolation in LUT's which contain these factors.

If blinds are present, the multipliers are scaled to reflect the FCT being computed: If the FCT is for the glass visible through the blinds, then each multiplier is factored by the fraction of glass visible from the target point. If the FCT is for the blinds, each multiplier is factored by the fraction of glass which is not visible.

Variables / Arrays

SHADO - interpolated body shadow at the target point
BETAB - interpolated background luminance BRDF factor at the target pt.
BETAT - interpolated target luminance BRDF factor at the target pt.
COSV,SINV - cosine, sine of the viewing direction at the target pt.
DOESI - (0 = don't compute ESI, 1 = do compute ESI)
NHEAD - pointer to list of potentially interfering partitions
NLINK - link field for list of potentially-interfering partitions
DMUL - 2 x 2 output array of multipliers -- these are for the nodes of 1 cell of the LUT
FGL - glazing code for the fenestration:
1 = clear, no blinds on window
2 = diffuse (shades, drapes, or glass)
3 = clear, portion of window visible through blinds
4 = blinds portion of window
IT - identifies arrays giving blinds charactertistics

A sawtooth structure always has three positive dimensions, as do clerestory and skylight structures which are elevated above the ceiling. By convention, FDIM(2,-) is always the width of a window; FDIM(1,-) is always 0 for a window.

The following arrays describe window barriers (refer to section 4.7.1.5 of the CEL-1 User's Guide for correspondence with input parameters):

BDIS - distance
BL(1,-,-) - limit 1
BL(1,-,-) - limit 2
BH - protrusion
BREF - reflectance

The following arrays describe shades (refer to section 4.7.1.1 of the CEL-1 User's Guide):

SHADED - pull-down depth
SHADET - transmittance

The following arrays describe drapes (refer to section 4.7.1.2 of the CEL-1 User's Guide):

DRAPET - transmittance
DRAPED(1,-) - distance 1
DRAPED(2,-) - distance 2

The following arrays describe window blinds; refer to section 4.7.1.3 of the CEL-1 User's Guide:

BLTYPE - (1=horizontal blinds, 2=vertical blinds)
BLTHK - thickness of one vane
BLW - width of one blind vane
BLS - spacing between blind vanes
BLANG - blinds opening angle
BLREF - reflectance of blinds

The following arrays describe a light shelf; refer to section r.7.1.4 of the User's Guide:

SHZ - distance beneath window edge
SHR - reflectance
SHY - protrusion distance

Note that each fenestration element generates an interior "insert" for purposes of the interiro flux transfer analysis. For a window or a skylight which has no well, the reflectance of the insert is simply the reflectance of the glazing. For a 3-dimensional fenestration structure, the reflectance is the effective reflectance (from subroutine RHOEF) of the cavity.

Buildings and ground surfaces generate the following arrays and variables:

XB,YB,ZB - (x,y,z) coordinates of each corner of the rectangular surface. Coordinates are relative to the room origin, with axes parallel to room walls. The first of the four corners is that nearest the origin. To reach the remaining three corners, view the the surface from the outside and move clockwise around its perimeter.

NORM - unit length outward normal from the surface, relative to the room coordinate axes.

STYPE - identifies the type of the surface:

1 = surface is vertical; its line projection upon the horizontal plane can be represented by

$y = Ax + B$ for constants A and B

2 = surface is vertical and lies in a plane parallel to room north; its x-coordinate is

$x = B$

3 = surface is horizontal; its z-coordinate is

$z = B$

A,B - these vectors locate the building and ground surfaces as in the above description of STYPE.

Ground inserts are treated in the same fashion as horizontal building surfaces. The ground itself is placed last in the list and given essentially infinite coordinate limits, so it will intercept any downward light ray which another surface has not previously intercepted.

4.41 FCTWTP

Purpose

FCTWTP computes a multiplier table from a fenestration rectangle to a target point. The sum of all such tables (one for each rectangle spanned by the fenestration map entry) constitutes the FCT for the map entry.

Method

First, a list of potentially-interfering partitions is constructed. Then the fenestration rectangle is discretized into component pieces. FCTTP is called for each piece and the resulting 2 x 2 output arrays are combined to form the table for the fenestration rectangle.

Variables / Arrays

DMUL - 2 x 2 array contains 4 output multipliers from FCTTP, visualized as follows on the LUT cell in question:

```
DMUL(2,1)     DMUL(2,2)
    +             +




    +             +
DMUL(1,1)     DMUL(1,2)
```

T - 21 x 21 x 4 output multiplier table. T(I,J,K) is the (I,J) element of the K^{th} plane, where K is:

1 - horizontal fc
2 - raw fc with body shadow
3 - background luminance
4 - target luminance

4.42 FENGET

Purpose

FENGET reads parameters describing fenestration from the user input deck. Arrays which describe the fenestration are constructed. Building surface parameters are also generated.

Method

The entire FENESTRATION block is read from the input deck. The following arrays and variables are set:

FTYPE - identifies the type of fenestration defined in each sub-block:
1 - window
2 - clerestory
3 - sawtooth
4 - skylight

FSHR - glaze type for each sub-block:
1 - clear
2 - diffusing

FTRANS - transmittance of each surface in the fenestration sub-block (up to 5 surfaces per fenestration sub-block definition):

Window - FTRANS(1,-) = transmittance, FTRANS(i,-) = 0 for i = 2,3,4,5

Clerestory - FTRANS(i,-) = transmittance of face i, with i =
1 west face
2 north face
3 east face
4 south face
5 top of structure

Note that for clerestory, FTRANS(5,-) must be 0

Sawtooth - FTRANS(i,-) = transmittance, where i is the transmitting face, as numbered above for a clerestory. FTRANS(j,-) = 0 for j = i

Skylight - same as clerestory, except FTRANS(5,-) must be positive.

FREF - interior reflectance of each surface in the fenestration structure, where the numbering scheme is as described for FTRANS. For a window, only FREF(1,-) can be non-zero.

FDIM - gives the three dimensions of each fenestration structure: FDIM(1,-) = x-dimension; FDIM(2,-) = y-dimension; FDIM(3,-) = z-dimension

4.43 FESI

Purpose

FESI computes ESI at a specified target point.

Method

The auxiliary file RCS866, which contains the luminance values vs. RCS (Relative Contrast Sensitivity), is read into memory at the first call to FESI. This file is used only for the computation of the inverse RCS; RCS itself comes from the subroutine RCS. ESI is computed as follows:

1) $$CRF = \frac{|L_b - L_t|}{L_b \; SC}$$

where

CRF = contrast rendering factor
L_b = background luminance
L_t = target luminance
SC = sphere contrast

2) RCS' = RCS x CRF

where

RCS = relative contrast sensitivity, a function of L_t
RCS' = adjusted RCS

3) L_e (equivalent sphere luminance) is then obtained from the RCS866 file via linear interpolation based on the RCS' value. If RCS' is less than the lowest tabulated value of RCS in the table, then an L_e is interpolated between (0,0) and that lowest tabulated value.

4) $ESI = L_e \times (E_t / L_t)$, where E_t is the raw illuminance with body shadow

4.44 FFPAR

Purpose

FFPAR computes the form factor between a pair of parallel surfaces. The form factor F_{ij} from surface i to surface j is the fraction of flux leaving i which is directly incident upon j.

Method

Refer to Figure 4.44a. The form factor from the source surface to the receiving surface is given by:

$$F = \frac{z^2}{2\pi A}\Big[b(1+a^2)^{\frac{1}{2}} \tan^{-1}\frac{b}{(1+a^2)^{\frac{1}{2}}} + \frac{b^2}{2}\ln\left(\frac{1+a^2+b^2}{b^2}\right) + a(1+b^2)^{\frac{1}{2}}\tan^{-1}\frac{a}{(1+b^2)^{\frac{1}{2}}} + \frac{a^2}{2}\ln\left(\frac{1+a^2+b^2}{a^2}\right) - \frac{1}{2}(1+a^2+b^2)\ln(1+a^2+b^2) + \frac{1}{2}(1+a^2+b^2)\Big]$$

evaluation limits: $\begin{bmatrix} i{:}2 & j{:}4 & k{:}2 & l{:}4 \\ i{:}1 & j{:}3 & k{:}1 & l{:}3 \end{bmatrix}$

$a = x_j - x_i$

$b = y_l - y_k$

A = area of source surface

Note that this is a solution to a four-dimensional integral and that the expression is to be evaluated a total of 16 times, once for each unique combination of the limits given with the angle brackets on the right. The terms are to be combined with appropriate sign changes as the integration limits change.

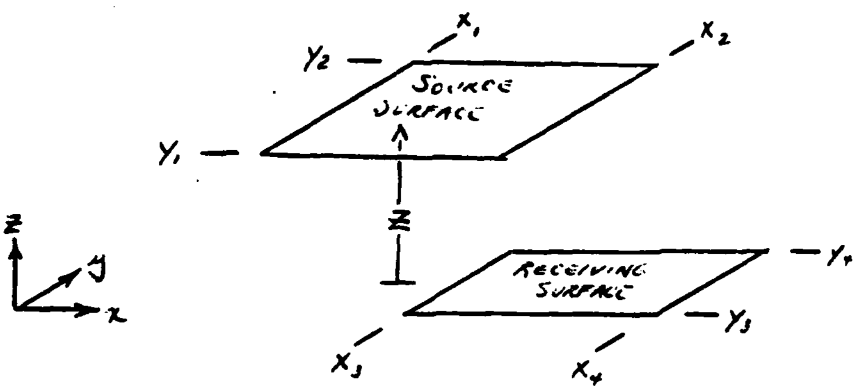


Figure 4.44.a: Parameters involved in evaluating form factor for parallel surfaces. z is distance between surfaces.

4.45 FFPER

Purpose

FFPER computes the form factor between a pair of surfaces which lie in planes which are perpendicular to one another.

Method

Refer to Figure 4.45.a. The form factor from the source surface to the receiving surface is given by:

$$F = \frac{-1}{2\pi A}\Big[a\,(x^2+b^2)^{\frac{1}{2}}\tan^{-1}\frac{a}{(x^2+b^2)^{\frac{1}{2}}} + \frac{a^2}{2}\ln\Big(1+\frac{x^2+b^2}{a^2}\Big) - \frac{1}{4}(a^2+b^2+x^2)\ln(a^2+b^2+x^2) + \frac{1}{4}(b^2+x^2)\ln(b^2+x^2) + \frac{1}{4}a^2\Big]\begin{bmatrix} i=2 & j=2 & k=4 & l=4 \\ i=1 & j=1 & k=3 & l=3 \end{bmatrix} \text{evaluation limits}$$

$a = y_j - y_k \qquad x = x_i - \bar{x} \qquad \bar{x}$ = x-coord. of receiving surface

$b = \bar{z} - z_l \qquad A$ = area of source surface $\qquad \bar{z}$ = z-coord. of source surface

Note that this is a solution to a four-dimensional integral and that the expression is to be evaluated a total of 16 times, once for each unique combination of the limits given with the angle brackets on the right. The terms are to be combined with appropriate sign changes as the integration limits change.

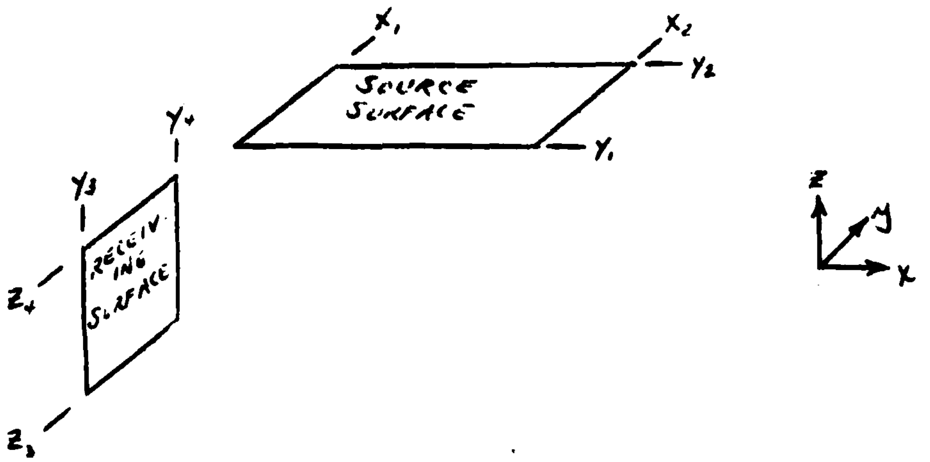


Figure 4.45.a: Parameters involved in evaluating form factor for perpendicular surfaces.

4.46 FILPRT

Purpose

FILPRT "completes" a prototype FCT. I.e., for given daylight conditions, FILPRT takes the destinations of the LUT rays from the fenestration and computes a table of luminance values at those ray destinations.

Method

Upon entry into the routine, the illuminance on building and ground surfaces due to the sky and sun will already have been calculated. If the window has barriers, this routine computes the final luminance on each barrier (see discussion of DR48).

1. For clear glazing:
For clear and partly skies, the routine must perform ray tracing to determine if direct sunlight can strike ray-destination points on a building or the ground. If a ray strikes the sky, the luminance is computed by the subroutine SKYLUM.

2. For diffuse glazing:
In this case the procedure is to compute the vertical illuminance outside the window and multiply it by the transmittance of the glazing (The 441 ray destinations are not used). The final luminance outside a window is the flux transfer solution for the barrier cavity; this derives from a 48-ray projection and is thus more reliable than a vertical illuminance calculation which ignores the sky-shadowing effect of other buildings. A ray-trace is performed to determine if the sun can strike the outside of the window. Note that the luminance value for diffuse glazing is used at each one of the 441 LUT points.

3. Blinds
The final luminance apparent on the blinds is computed by the routine BLILUM. As the blinds are considered Lambertian, the 441 LUT-table values are all set to this final luminance.

Variables / Arrays

IMAP - identifies the fenestration map entry whose prototype FCT is to be computed
NS,SREF,STYPE,XB,YB,ZB,A,B,NORM - describe the building surfaces (see discussion of FENGET)
ISKY - the sky condition (1=overcast, 2=clear, 3=partly cloudy)
IPOLL - not used in this implementation. This parameter may be used to identify atmospheric conditions in future enhancements
CAND - output 21 x 21 luminance table at the LUT points
LUPROT - logical unit # of disc file containing the ray destinations
LUB48 - logical unit # of disc file containing the ray destinations from window barriers and the flux transfer matrix for the barrier cavity

HSURF, COSIN, HXYZ, AZIM, ELEV - arrays which describe the projected rays' destinations.

SFCSKY - 3 x 30 array gives the illuminance due to the unobstructed sky on building and ground surfaces.

SFCSUN - vector which gives the solar illuminance on building and ground surfaces.

ZL - zenith luminance

MISUN - index of the lower row in the FCT which contains a solar component.

MJSUN - index of the leftmost column in the FCT which contains a solar component.

SUNCO - coefficients (either 1 or 0) indicating whether or not the sun is visible at the 4 FCT points ((MISUN,MJSUN), (MISUN+1,MJSUN), (MISUN,MJSUN+1), (MISUN+1,MJSUN+1))

SUNM - the weights at the 4 FCT points which divide the solar component

4.47 FSTOTP

Purpose

FSTOTP computes the FCT from a given fenestration map entry to a target point.

Method

The list of all fenestration source elements is searched. Each element which "belongs" to the map entry is discretized into pieces no greater than 5' on a side (subroutine DISCW). DISCW then calls FCTWTP to compute a multiplier table for each of the (5' or less) fenestration pieces. The sum of all these multiplier tables for all (5' or less) pieces for all fenestration rectangles constitute the required FCT.

4.48 FTAPAR

Purpose

FTAPAR computes, via flux transfer analysis, the final average luminance on each interior surface. (This routine is not used by the LUMEN-II subset of programs)

Method

The routine reads the flux transfer matrix from disc, then loops thru all the interior surfaces and reads their point-by-point illuminance calculation results. The average illuminance on each surface is computed, and then multiplied by the negative of the reflectance to yield the right-hand-side vector of the flux transfer formulation. The matrix solution yielding the final average luminances (AVLUM) is determined by the subroutine GSEIDL.

4.49 FTPDSK

Purpose

This routine is the "driver" subroutine for the computation of all FCTs for all target points. The routine computes one FCT per target point for each fenestration map entry. The FCTs are saved on disc for later access.

Method

The outer loop is on fenestration map entries; the inner loop is on target points. A FCT is computed (in subroutine FSTOTP) for each combination of map entry and target point.

Note that each FCT disc record for target points contains 1768 words, laid out as follows:

words	
1 - 441	FCT for horizontal illuminance
442- 883	FCT for raw fc with body shadow
884-1325	FCT for background luminance
1326-1764	FCT for target luminance
1765-1766	lower and upper row indices specifying the smallest subset of the FCT which contains all the FCT's nonzero entries.
1767-1768	leftmost and rightmost column indices specifying the smallest subset of the FCT which contains all the FCT's nonzero entries.

Variables / Arrays

TORG3 - height of target plane above floor
LUFCTP - logical unit # of disc file which is to receive the FCT output tables.
NMAP - number of fenestration map entries
DOESI - (0 = don't calculate ESI; 1 = do calculate ESI)

4.50 GSEIDL

Purpose

GSEIDL solves a system of linear equations using Gauss-Seidel iteration.

Method

Let the system of equations be:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2$$

$$\vdots$$

$$a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n$$

Assume that $a_{ii} = 0$ for all i (any solvable system may be written in this form; our flux transfer matrix is already in this form, with each $a_{ii} = -1$). Then we solve each equation for the x_j on the diagonal:

$$1) \qquad x_i = (1/a_{ii})\,(\,b_i - a_{i1}x_1 - a_{i2}x_2 - \dots - a_{in}x_n)$$

We start with initial guess $x_k = b_k$ for all k and then solve each equation 1) for x_i, using the most current estimate for all the x_j, $j \neq i$. The algorithm terminates when either:

a) the preset maximum # of iterations has been performed
or
b) the largest change in consecutive approximations of any x_i is less than a specified threshold.

Variables / Arrays

N - dimension of the system
A - matrix of coefficients
B - right-hand side vector
MAXIT - maximum # iterations to be performed before terminating
RMAXCH - threshold permissible change in any x_i between consecutive iterations. The algorithm terminates when none of the x_i change as much as RMAXCH on consecutive iterations.
NN - the dimension of A in the calling program

4.51 GSET

Purpose

GSET computes the "new" luminaire gain settings from the "current" settings and the current controlling illuminance value. This routine applies to "on-off" and "high-low-off" controls only.

Method

The coding is a straightforward implementation of the tables shown on pp. 127-128 of the CEL-1 User's Guide.

Variables / Arrays

VAL - the current controlling illuminance value

KG - a 3 x 5 array (used for high-low-off controls) whose entries yield the next gain setting. The rows are indexed by the current gain (1=off, 2=low, 3=high); the columns are indexed by the range of the current controlling illuminance value.

4.52 HFCSKY

Purpose

This routine computes the horizontal fc on the earth due to the unobstructed sky.

Method[8]

The computation employs parameters A, B, and C in the following expression:

$$E = A + B \sin^C h$$

where h is the solar altitude and A, B, and C are:

	A	B	C
overcast sky:	27.88	1952	1.0
clear sky:	74.35	1441	0.5
partly cloudy sky:	27.88	4182	1.0

Note that the expression for E above yields a result in footcandles.

4.53 HFCSUN

Purpose

HFCSUN computes horizontal fc on the earth due to direct sunlight.

Method[8]

The expression for solar illuminance is:

$$E = 11849 \left(1 + .033 \cos \frac{360\ J}{365}\right) e^{-a/\sin H} \sin H$$

where

J is the Julian day

H is solar elevation

a varies with sky condition:

a = 0.80 partly cloudy skies

= 0.21 clear skies

4.54 IIOS

Purpose

IIOS computes illumination point-by-point on the x-y plane from a given luminaire location. The mounting height is always 1', so an adjustment factor is applied to account for the actual mounting height. The argument list describes a list of potentially interfering partitions, and their shadowing effect is accounted for.

Method

For each point on the x-y plane the luminaire is discretized according to the argument PARM. Discretization is such that the largest luminaire piece is no greater in either dimension than (1./PARM) times the distance from the luminaire center to the calculation point. A ray trace is performed from each luminaire piece to the calculation point. If any partitions block the view, then the contribution is zero; otherwise a value is interpolated from the point source LUT and factored according to the actual distance from the luminaire to the calculation plane and the degree of luminaire discretization.

Variables / Arrays

XLEFT, YBOT - the x and y-coordinates of the point on the calculation plane which is nearest the origin
DELX - distance between columns on the calculation grid
DELY - distance between rows on the calculation grid
NR - # rows on the calculation grid
NC - # columns on the calculation grid
XDIM, YDIM - x and y dimensions of the luminaire
ZSQ - the square of the actual mounting height of the luminaire above the calculation plane
FC - contains accumulated illuminance at the calculation points
T - contains the point source LUT
HEAD, LINK, DIR, X1, X2, Y1, Y2, Z1, Z2- define the linked-list structure which describes the interior partition surfaces
GC, C, NV, JQCAL - dummy parameters not used in this implementation
PARM - governs the degree of discretization performed on the luminaire. Refer to the "Method" section for calculation details.

4.55 INDTPS

Purpose

INDTPS computes the interreflected contribution from the interior surfaces to target points, area points, and sensors.

Method

The routine uses the destinations of the 48-ray projections from each target point (see discussion of subroutine D048), the final luminance on each interior surface, and a table of multipliers (computed in subroutine BRDF) to determine the illuminance at the target points. For area points and sensors, the multipliers are each 1/48.

Variables / Arrays

HEAD, LINK, C1, C2, Xi, X2, Y1, Y2, Z1, Z2 - describe the list of interior partition surfaces (these parameters are not used)
AVLUM - final luminance on each interior surface
VIEWD - the viewing direction at each target location
FC - contains the direct component illuminance on the target points
RFC - contains the raw fc w/body shadow at the target points (direct comp.)
LB - contains the background luminance (direct component) at target pts.
LT - contains the target luminance (direct component) at the target pts.
SFC - contains direct component illuminance on the interior sensors
LU48 - logical unit # where the destinations of the 48-ray projections are stored
CLUM, ICELNG - not used
APFC - contains direct component illuminance on area points.
ZONEF - 48-ray projection multipliers for horiz. fc (= 1/48 everywhere)
ZONER - 48-ray projection multipliers for raw fc w/body shadow
ZONEB - 48-ray projection multipliers for background luminance
ZONET - 48-ray projection multipliers for target luminance

4.56 KNTOU (also LNTOU)

Purpose

KNTOU is used in a partitioned and/or daylight environment to convert a linear list of target points into a rectangular grid. This is done so that the target points can be displayed as a rectangular grid. KNTOU is invoked because, in a partitioned and/or daylight environment, UNKNOWN task locations (rectangular grid) are converted to a linear linked list before computations are performed. They must be converted back to a grid before being printed out. Recall also that individual target point values are not printed out in energy profile mode, so that KNTOU/LNTOU are not invoked in energy profile mode.

The distinction between KNTOU and LNTOU is that KNTOU is used for ESI-related quantities (i.e., those where a viewing direction applies); LNTOU is used for horizontal illuminance only.

Variables / Arrays

V - linear array of computed values at each target location

VIEWD - viewing direction at each target location

XTP,YTP - (x,y) coordinates of target locations

A - rectangular array of computed values (output)

4.57 LGAINS

Purpose

In the energy profile calculations, groups of luminaires which are controlled alike are treated as one "quasi-luminaire." Hence the gain applies to an entire group of luminaires. LGAINS allows the calling program to determine the gain of each individual luminaire.

Method

The array LGRP defines the groups of luminaires which are dimmed together. I.e., LGRP(I) = 0 if luminaire I is always off
= the dimming group # if luminaire is not always off

This means that if LGRP(I) = LGRP(J), then luminaires I and J are controlled together and must always have the same gain. The array GG contains the gains of each luminaire group, so the gain of luminaire I is

GG(LGRP(I))

4.58 MAPAR

Purpose

MAPAR accumulates the contribution from a given luminaire to a given partition surface. The argument list describes any potentially interfering partitions, and their effect is accounted for.

Method

A coordinate transformation is undertaken which maps the target surface onto the x-y plane; the same transformation is applied to the luminaire and to the list of potentially-interfering partitions. The previously-accumulated illumination at the surface (point-by-point) is read from disc, along with the appropriate point-source LUT. The subroutine IIOS is then called to perform the point-by-point calculations. Finally, the newly-accumulated contribution is written back to disc.

Variables / Arrays

LUR - logical unit # where point-by-point accumulations reside
LUT - logical unit # of the disc file where the point source LUT's reside
TPL - integer which identifies the surface for which point-by-point calculations are to be performed
NR - # of rows of points on the surface
NC - # of columns of points on the surface
HEAD, LINK, C1, C2, DIR - define the linked list of potentially-interfering partitions
XEW, YNS - x,y dimensions of the luminaire
XL, YL, ZL - (x,y,z) coordinates of the luminaire location
FC - array used to accumulate the contribution
T - array used to contain the point-source LUT.

4.59 METENG

Purpose

METENG converts metric distance measurements to feet.

Method

The calling program passes 5 vectors of real numbers (meters), plus the length of each vector. METENG converts all elements of each vector to feet:

$$\text{dimension in feet} = 3.2808 \times \text{dimension in meters}$$

Note that the calling program may convert a scalar variable by passing its dimension as 1.

4.60 MSTRSN

Purpose

This routine is used in daylighting calculations. It is called once for each different daylight condition; it computes the gain setting for each luminaire and the consequent energy consumption.

Method

I. Analysis Mode

In analysis mode, all luminaires are operated at gain = 1. Therefore all that is necessary is to add the contribution of each luminaire to the daylight contribution.

II. Profile Mode - Continuous Dimming

1. Accumulation arrays are initialized to contian the daylight contribution.

2. All luminaire gains are set to 1.

3. The contribution from each luminaire is added to the accumulation arrays from 1.

4. Let: DBAND(1) be the criterion illuminance which must be maintained
   FCRIT(J) be the illuminance obtained from the system when the gain of luminaire J is tentatively reduced by an amount DQ, with each other luminaire holding its previous gain

5. If this is the first time through this step, set DQ = 0; otherwise, set DQ = 0.1
   For each luminaire J, tentatively reduce its gain by DQ and compute FCRIT(J).

6. Set FCMAX = max FCRIT(J), J=1, ... ,# luminaires

   set JMAX = the luminaire for which FCMAX = FCRIT(J)

   If FCMAX is less than DBAND(1), then the luminaires cannot be dimmed further.

7. Permanently reduce the gain of luminaire JMAX by DQ. Adjust the accumulation arrays accordingly and return to step 5.

In other words, each step of the process involves reducing the gain of one luminaire by the fraction DQ. The luminaire which is selected is the one which leaves the largest gap between the resulting actual illuminance and the criterion value.

III. Profile Mode - High-low-off or On-off controls

1. Accumulation arrays are initialized to contain the daylight contribution.

2. Luminaire gains are set to the values passed from the calling program (at the first time instant of the day, gains should all be zero; at other times, gains will be what they were set to at the previous time on the same day).

3. The luminaire contributions (with gains from step 2) are added to the daylight contribution in the accumulation arrays.

4. The subroutine GSET is called to compute the "new" gain settings.

Note that the term "luminaire" as used above refers to "quasi-luminaire", where all luminaires which are controlled together are treated as a single "quasi-luminaire." This reduces the computer time required to compute the new gain settings, especially for the continuous dimming case.

Variables/ Arrays

LUCONT - logical unit # of disc file containing contribution of each quasi-luminaire
ESI - ESI values at the target points
ESIR - ESI Ratings at the task locations
CGAIN - the "current" gain settings (input parameter)
QW - the quadratic coefficients giving watts vs. gain
TWATT - total watts consumed (output)
NQLUM - number of quasi-luminaires
GL - new gain settings (computed output)
INDW - identifies the quadratic coefficients associated with each luminaire
NCONTR - the number of quasi-luminaires which are to be controlled. This leaves NQLUM-NCONTR luminaires which are always operated at full gain. For these latter luminaires, one "quasi-luminaire" = one actual luminaire
HFC,TE,BL,TL,APFC,SENFC - accumulation arrays for horizontal fc, raw fc w/body shadow, background luminance, task luminance horiz fc at area points, and illuminance at the interior sensors, respectively

FC,ET,LB,LT,AFC,SFC -- same as HFC,TE,... , except these contain the contribution from one "quasi-luminaire"

4.61 OBSF

Purpose

OBSF opens four binary scratch files needed to hold intermediate calculations. Under input parameter control, each file may be initialized to all zeroes.

Method

The integer array FILES contains up to 15 logical unit numbers. A scratch file is opened for each non-zero logical unit # in FILES (until the first zero is encountered). Each scratch file has 31 records of 441 words each.

4.62 OBSURF

Purpose

OBSURF breaks an obstruction within the room into it's component surfaces and adds each component surface to the partition list. The routine also computes the vertical obstruction area which lies beneath the target plane and the reflectance-weighted area beneath the target plane. Also, the horizontal obstruction surface area on the target plane is computed.

Method

From an obstruction location and dimensions, it is a straightforward matter to compute the coordinate limits and facing direction for any face on the obstruction. The following provisos apply:

a) Any horizontal dimension less than 0.335' causes any surface having that dimension to be ignored.
b) Any vertical dimension less than 0.667' causes any surface having that dimension to be ignored.

Variables / Arrays

HEAD, LINK, RHOS, X1, X2, Y1 , Y2, Z1, Z2 - define the linked list structure which describes the partitions.
WID, LEN, HT - width, length, height of the obstruction
ROT - rotation angle of the obstruction
REFL - Vector containing the 6 reflectances of the obstruction.
XC, YC, ZC - (x,y,z) coordinates of the top center of the obstruction.
TORG3 - z-coordinate of the target plane
AVAIL - pointer to the first cell in the linked-list available pool.
ABELOW - the area of vertical obstruction surface which lies beneath the target plane.
ABREF - a weighted sum computed by multiplying the area of each obstruction face which lies beneath the task plane by its reflectance.
ADESKS - four times the area of any horizontal obstruction surface which lies on or within 0.67 feet above the target plane.
ADREF - a weighted sum computed by multiplying each horizontal obstruction surface (which lies on or within 0.67' above the target plane) by its reflectance.

4.63 OPTMZ (also GRADE)

Purpose

OPTMZ is used in design synthesis. The optimum set of luminaires to be operated is determined. GRADE computes the difference between the design criteria and the illuminance obtained from a specified operating set of luminaires.

Method

The algorithm is similar to that used for continuous dimming in the subroutine MSTRSN, one difference being that the smallest permitted gain adjustment is 1 - i.e., each luminaire must be either off or operated at full gain.

1. All luminaires are turned on. If the design criteria are not met, the required operating subset has been determined.

2. A pass is made through all luminaires. If a luminaire has not previously been turned off, it is tentatively turned off here. Illuminance is computed with the one tentatively "off" luminaire and all luminaires which have been permanently turned off in a previous step not operating.

3. When all luminaires have been tentatively turned off one-by-one in step 2, that luminaire is permanently turned off which leaves the largest gap between the illuminance values and the design criteria.

4. Steps 2 and 3 are repeated until no luminaire can be turned off without violating the design criteria.

Subroutine GRADE is used in step 2 to compute the gap which tentatively extinguishing a luminaire leaves between computed illuminance values and the design criteria.

Note that OPTMZ is used also in non-synthesis mode, by calling it with the parameter IDESGN set to zero.

Variables / Arrays

AFC - permanent accumulation array for horiz. fc at the target pts
AET - permanent accumulation array for raw fc w/body shadow at the target pts
ALB - permanent accumulation array for background luminance at the target pts
ALT - permanent accumulation array for target luminance at the target pts
ESI - permanent accumulation array for ESI at the target pts
ESIR - permanent accumulation array for ESI rating at the task locations
XL,YL,ZL - (x,y,z) coordinates of luminaires
LUOUT - logical unit # for printed output
NLUMS - number of luminaires
IDESGN - (0=compute illuminance with all luminaires operating - no synthesis)
(1=compute optimum operating luminaire set)

DOESI - (0 = don't compute ESI;   1 = do compute ESI)
DMESI - ESI design criterion value
DMILL - minimum illuminance criterion value
DAILL - average illuminance criterion value

4.64 OSCU (also CEVAL, SLOPE)

Purpose

OSCU computes the osculating polynomial fit over a set of (x,y) points. The osculating polynomial is a piecewise cubic fit, where a different cubic may be used over each interval. However, the osculating polynomial has a continuous first derivative over its entire range.

SLOPE is called from OSCU to compute a slope at each tabulated point. CEVAL evaluates the function fitted with the osculating polynomial at a given point on the x-axis.

Method

Four coefficients determine a cubic polynomial over any given interval (x_i, x_{i+1}):

1) $y = c_0 + c_1 x + c_2 x^2 + c_3 x^3$

If we translate the interval so that its left boundary becomes the origin, the interval becomes $(0, x_{i+1}-x_i)$ and the evaluation becomes

2) $y = c_0 + c_1(x-x_i) + c_2(x-x_i)^2 + c_3(x-x_i)^3$

We then have $c_0 = y_i$, leaving only c_1, c_2, and c_3 to be determined. We differentiate 2) to obtain:

3) $y' = c_1 x + 2c_2(x-x_i) + 3c_3(x-x_i)^2$

c_1, c_2, and c_3 are then obtained by solving the system:

$$y_{i+1} = y_i + c_1(x_{i+1}-x_i) + c_2(x_{i+1}-x_i)^2 + c_3(x_{i+1}-x_i)^3$$

$$y_i' = c_1 x_i$$

$$y_{i+1}' = c_1 x_i + 2c_2(x_{i+1}-x_i) + 3c_3(x_{i+1}-x_i)^2$$

y_i' and y_{i+1}' are obtained from subroutine SLOPE. SLOPE computes y_k' by fitting a quadratic polynomial through (x_k, y_k) and the two points on either side; the slope at x_k is that obtained from the quadratic fit.

CEVAL evaluates the osculating polynomial at a given input point w by first locating the interval w lies in, then evaluating 2), with x = w.

4.65 PHOTO

Purpose

PHOTO reads photometric candela values from a file. The routine also computes the indexing arrays TNDX and PNDX which facilitate accessing candela values in the photometric array.

Method

The photometric data is read from logical unit # LU and factored by the lamp lumens (LUMENS) and light loss factor (LLF) supplied by the calling program. The variable KUD is set as follows:

KUD = 1 luminaire is downlight only
2 luminaire is uplight only
3 luminaire is both uplight and downlight

The variable KSYM is set as follows:

KSYM = 1 -luminaire distribution is symmetric about both the 0° and 180° photometric planes, and the 90° and 270° planes.
2 -luminaire distribution is symmetric about only the 0° and 180° photometric planes
3 -the luminaire distribution is completely asymmetric

Note that regardless of the luminaire symmetry, the candela table CD is expanded to span the entire sphere surrounding the luminaire.

The arrays TNDX and PNDX are used to avoid a linear search through the angles to find the cell in CD over which we should interpolate to find a candela value at a given lateral and vertical angle. Accordingly, TNDX and PNDX locate the desired interval, provided the THETA and PHI arrays do not contain any consecutive angles closer than $\frac{1}{2}^{\circ}$. TNDX contains an index value for each $\frac{1}{2}^{\circ}$ vertical angle interval between 0° and 180°. For example, TNDX(1) locates the interval in the THETA array where any angle in the range $(0,\frac{1}{2})$ lies; TNDX(2) locates the interval where any angle in the range $(\frac{1}{2},1)$ lies; etc. For example, suppose that there are 12 THETA (=vertical) angles:

0, 5, 15, 25, 35, 45, 55, 65, 75, 85, 90, 180 degrees

Then TNDX(1) thru TNDX(10) all contain the value 1 (which tells us we should interpolate in the first interval, 0-5 degrees); TNDX(11) thru TNDX(30) all contain 2 (indicating the second interval, 5-15 degrees); etc.

PNDX is determined for the PHI (lateral) angles in precisely the same fashion. Note that the PNDX array, however, spans 0-360 degrees.

4.66 PLOTS (also PVALS, PLUG, TCELLS, CONTUR, CUPOL, PSAL)

Purpose

These subroutines generate the character contour plots.

Method

I. PLOTS

First, PVALS is called to determine the 5 values for which to draw contours. PVALS determines the minimum and maximum over the array of numbers and divides this interval into n+1 subintervals, where n is the number of contour values to plot. The plot values are then the n interior points dividing the subintervals.

The plotting area and scale are determined (subroutine CHROOM). The plotting area is initialized to all blanks, except for + signs spaced 1" apart in both directions; the room boundaries are drawn in on the perimeter of the plot area.

CONTUR is called to generate a set of (x,y) coordinates which constitute a reasonably closely-spaced sample of points along the contour. Each call to CONTUR results in either a set of contour points or a report that all curves at that value have already been generated.

The (x,y) coordinates along a contour are passed to PLUG, where the character which represents this particular contour is substituted into the appropriate positions in the character map of the room. When this has been done for all contours, the plot can be printed.

NVALS - # of values to be plotted (input parameter). If zero, PVALS is called to select 5 plot values.
CHART - array of characters which represent the room. Characters are substituted for blanks as the contours are generated.
LPI - lines per inch on printer (up-down).
CPI - characters per inch on printer (across).
DELC - column spacing of target point array.
DELR - row spacing of target point array.
ORG - (x,y) coordinates of target point nearest the origin.
LU - logical unit # for printed output.

II. CONTUR

CONTUR operates in essentially a two-phase process:

1. A "coarse" set of points on the contour is located. Each of these points lies on a row or column of the target grid.

2. A "fine" set of points along the contour is obtained by fitting smooth curves through the coarse set.

The coarse set of points is isolated by the subroutine TCELLS. The fine set is obtained by parameterizing the x and y points of the coarse set separately, and then fitting an osculating polynomial

through each parametric representation. The parameterization is in terms of "pseudo arc length" -- i.e., the distance obtained in traveling straight-line paths between points on the coarse set.

An arc-length function A is tabulated at n points, the number of points in the coarse set, where A_i for any i is the cumulative distance traversed in straight line fashion from (X_1,Y_1) to (X_2,Y_2) to ... to (X_i,Y_i). I.e., our arc-length function is defined:

$$A_1 = 0$$

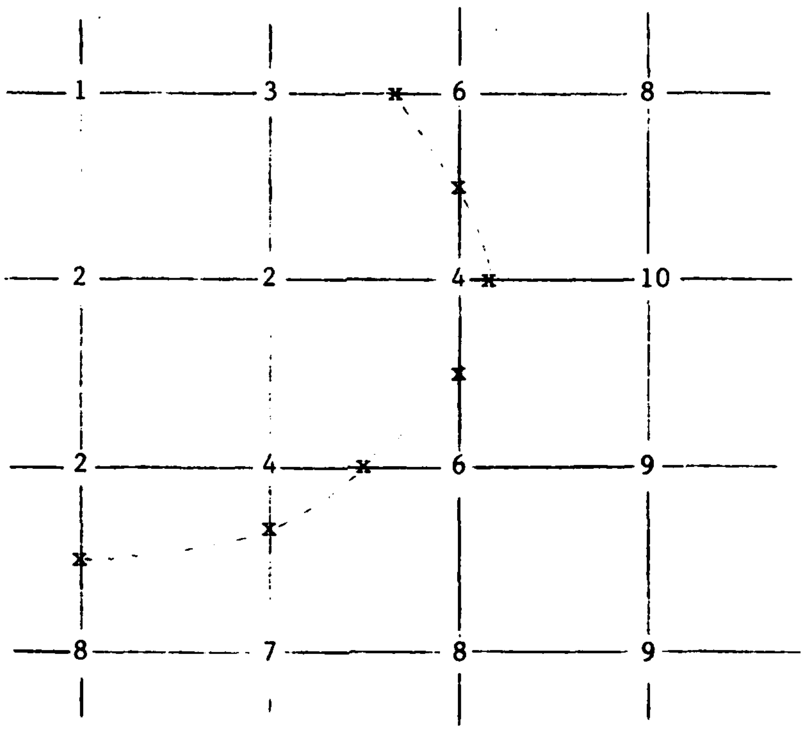
$$A_i = A_{i-1} + \left((X_i-X_{i-1})^2 + (Y_i-Y_{i-1})^2 \right)^{\frac{1}{2}} \text{ for } i=2, \ldots ,n$$

Two calls are then made to OSCU, the first with (A,X) as the independent and dependent variables, respectively; the second with (A,Y) as the independent and dependent variables. The resulting fits will permit us to obtain as fine a spacing as desired on the contour.

T - input 2-dimensional array of numbers over which contours are desired.
NROWS,NCOLS - number of rows and columns in T
X,Y - "fine" set of (x,y) points on the contour (output)
N - the number of (x,y) coordinate points on the contour
DEL - stepsize for determining the (x,y) set
TAG - auxiliary array used to avoid retracing the same contour
VAL - value for which a contour is to be determined
ICON - assumed equal to 2
DELC,DELR - grid column and row spacing, respectively.
ORG - (x,y) coordinates of the grid point nearest the origin
U,V - (x,y) coordinates of the "coarse" set
A - pseudo arc length along the coarse set

III. TCELLS

TCELLS uses linear interpolation on grid rows and columns to determine where a contour line crosses them. For example, suppose that the numbers below represent the values at nodes on a portion of the grid and that we want a contour line for 5 fc. The x's in the sketch below roughly represent the coarse set chosen :



In other words, TCELLS tracks a contour from cell to cell in the input grid and interpolates to determine exactly where the contour enters and leaves cells.

IV. PLUG

PLUG translates a set of (x,y) coordinates in the room to the corresponding positions in the character map of the room. The given character (which represents the contour in question) is substituted at each position along the contour.

V. PVALS selects plot values as described in I.

VI. PSAL computes the pseudo arc length as described in II.

VII. CUPOL evaluates a cubic polynomial for a set X of input points. It is identical to the CEVAL subroutine (described in 4.64) except that CEVAL only evaluates the polynomial at one input point per call.

4.67 PROFCT

Purpose

PROFCT computes a prototype FCT for a given fenestration map entry.

Method

The prototype FCT describes the destination of rays projected from fenestration along the 441 LUT points. In every case the actual (x,y,z) location from which the rays are projected are the center of the room surface for which the map entry applies. The prototype FCT contains the following information:

a) Ray strikes sky:
   AZIM - azimuth of sky point struck, relative to room north
   ELEV - elevation of sky point struck
   (Note that AZIM and ELEV are EQUIVALENCED to the first 2 planes of HXYZ)
   HSURF - set to 0 to indicate sky was struck

b) Ray strikes ground or building surface:
   HXYZ - (x,y,z) coordinates of point struck
   HSURF - set to the number of the surface struck (positive integer)
   COSIN - cosine of angle between the ray and normal to the surface

c) Ray strikes window barrier, or interior surface of clerestory, skylight, or sawtooth structure:

   In this case all is the same as in case b), except that HSURF is set to the negative of the barrier or interior surface number. Barriers are numbered as in the User's Guide. Interior surfaces are numbered in the same fashion as the room surfaces -- 1=west face, 2=north face, 3=east face, 4=south face, etc.

4.68 PROFIL

Purpose

PROFIL prints the energy profile for a given month.

Method

The subroutine BGRAH formats the profile data for printing. PROFIL computes/uses the following parameters:

PE - 24 x 4 array giving the fraction of total possible energy consumption as a function of hour and sky condition. The fourth column of the array is the mean expected consumption vs. the hour

KWH - 24 x 3 array giving the power requirements (watts) as a function of hour and sky condition. Since each value represents one hour, it may be thought of also as watt-hours.

PROB - 24 x 3 array giving the probability of each sky condition prevailing; these are given for each hour, but in the present implementation these probabilities do not vary from hour to hour. E.g., the probability that the skies will be overcast is simply the number of overcast days in the month divided by the total number of days in the month, and this probability holds for each hour.

YAG - yearly aggregate energy consumption vs. the number of working days per week.

LUSKED - the logical unit number of the disc file where the energy consumption schedule is to be written. LUSKED=0 implies no schedules are to be written to disc. Schedules are written to disc for subsequent processing by CEL07 (the BLAST interface program).

4.69 PROJ48

Purpose

PROJ48 projects the 48 rays from a target point in an interior environment (see section 2.3). The destinations of the rays are saved on disc for subsequent retrieval and processing.

Method

First, the lists of vertical partitions are sorted into the following orders:

1. west facing - ascending order
2. north facing - descending order
3. east facing -descending order
4. south facing - ascending order

The lists are then pruned so that only potentially-interfering partitions remain in the list; e.g., the west-facing list if pruned of all partitions which lie west of the target point.

The 48 rays are then generated, one apiece for each combination of azimuth angles (15, 45, ... , 315 degrees) and elevation angles (15, 37½, 52½, and 75 degrees). The first partition running east-west and the first partition running north-south which the ray strikes are determined. A distance comparison is performed to determine which of these 2 surfaces the ray strikes first. If no east-west or north-south running partition is struck, then the ray must strike the ceiling. IN this latter case we must check to see if any ceiling fenestration openings occupy the point of strike. If so, the destination surface # is set to the "insert" number which corresponds to the ceiling fenestration.

Note that if the ray destination is the ceiling or any of the walls, an artificial ray destination # is generated which locates the particular discretization zone of the surface struck. Thus, the indirect component calculation treats each room surface zone as if were a "surface" by itself with its own distinct luminance.

4.70 QLUM

Purpose

For energy profile calculations, it is advantageous to treat all luminaires which are controlled alike (i.e., must always have the same gain) as a single "quasi-luminaire." This routine computes the contribution of each "quasi-luminaire" from the component contributions of the actual luminaires which make up the "quasi."

Method

The number of "quasi-luminaires" is equal to the number of dimming groups (limited to one in this implementation; all luminaires within one dimming group must have the same gain) plus the number of luminaires which are always on.

The array Q is used to accumulate the contributions from all luminaires within each dimming group; each such aggregate contribution then becomes the contribution from the corresponding "quasi-luminaire."

4.71 RASTRK

Purpose

RASTRK traces a ray traveling outside the room; it returns the building or ground surface number where the ray hits, together with the (x,y,z) coordinates of the strike point and cosine of the angle between the ray and a normal to the surface. Alternatively, the routine returns an indication that the ray strikes the sky.

Method

Let $\vec{p} = (p_x, p_y, p_z)$ be the origin of the ray (often the center of a window, for example). Let $\vec{q} = (q_x, q_y, q_z)$ be the vector of direction cosines which describe the ray's direction (this will often be the direction obtained by projecting a ray from a LUT point through the LUT origin). Any point $\vec{r}$ on the ray may then be written:

$$\vec{r} = \vec{p} + \lambda q \quad \text{for some real } \lambda$$

Each building/ground surface is considered in turn. First the cosine between the projected ray and the unit outward normal from the surface is computed:

$$\text{cosine} = q_x n_x + q_y n_y + q_z n_z$$

This cosine must be either positive or negative (according to an input parameter) for a "hit" to be possible. If the sign is right, then the (x,y,z) coordinates of the strike point on the surface are computed. For example, if the surface is horizontal, with its z-coordinate = B, then we solve for :

$$\lambda = \frac{(B-p_z)}{q_z}$$

then

$$x = p_x + \lambda q_x$$

$$y = p_y + \lambda q_y$$

x and y are compared to the coordinate limits of the surface to see if the point lies on it.

If all surfaces are examined, and the ray strikes none of them, then it must strike the sky.

4.72 RCS

Purpose

RCS computes Relative Contrast Sensitivity as a function of target luminance (given in footlamberts).

Method

The input luminance is converted to nits (metric). Then one of 8 cubic polynomial representations is chosen, the the RCS is evaluated using the four cubic coefficients selected. The representation is chosen by dividing the input argument by the square root of 10 until the quotient lies between 0 and $\sqrt{10}$.

4.73 RSDET (also RELFC, ICOVER, VWTOSP, HSTOTP)

Purpose

Recall that the FCT scheme for interior surfaces stores an FCT which corresponds to the average illuminance on a given surface. Thus, the FCT scheme does not reveal the gradient on interior surfaces. For obstructions within the room, whose surfaces figure to be reasonably small, treating each face as uniformly luminous should lead to no serious compromises. However, the ceiling and walls may exhibit gradients which it is essential to model more carefully. These subroutines compute an estimate of the relative initial illuminance (due to daylight) on the walls and ceiling. These relative weights can be used after the FCT evaluation has been made to adjust the point-by-point luminance in a manner consistent with the average computed under the FCT scheme.

Method

An illuminance value is computed at the center of each discrete zone on the walls and ceiling. This illuminance value is based on a uniformly bright sky; since it will be used for all sky conditions, this seems a reasonable approximation.

A loop is taken through all fenestration, each fenestration rectangle is descretized, and the contributions are added up. 5 records (one for each of the 4 walls, plus the ceiling) are written to disc file LUSLUM; these records contain the computed illuminance values (in the relative sense only - no attempt is made to emerge with foot-candles) at the center of each zone.

4.74 RHOEF

Purpose

RHOEF computes the effective reflectance at the open face of a rectangular 3-dimensional cavity. This calculation is useful in a partitioned environment in that the cavity beneath the target points may be ignored, provided we compute the effective reflectance of the target plane.

Method

The algorithm[7] is based on traditional flux transfer analysis, where the "surface" at the open face has zero reflectance and 100% of the initial illuminance. Motivation and other details may be found in the original reference.

4.75 RSLTS

Purpose

RSLTS prints the outcome of a daylight calculation for one given time.

Method

The coding is straightforward. The following points are worth noting:

1. A legend which numbers each target point according to its (x,y) location is printed the first time this routine is called, unless we're in analysis mode with UNKNOWN task locations (rectangular grid of target points).

2. In profile mode, only a statistical summary of target point calculations is printed.

4.76 SANDB

Purpose

SANDB is useful when it is possible to predict that several consecutive interpolations will be required between the same 2 rows in the LUT. This happens, for example, when computing luminaire contribution to a row of target points. The row of target points will all be offset at a given y-distance into the LUT (I.e., between 2 given rows in the LUT). Foreknowledge of these 2 rows enables us to precalculate coefficients which facilitate the subsequent interpolations.

Method

Suppose that we are preparing to interpolate at a given row-distance y in the LUT. We first locate the LUT rows which bound y:

$$i = \mathrm{INDX}(4y+41)$$

See section 2.1 for a discussion of the INDX array. Now rows i and i+1 surround y in the LUT. We compute slope and intercept coefficients s and b which yield an interpolated value at (x,y) in the LUT:

$$\text{Value} = sx + b$$

A set (s_k, b_k) are computed for every interval k in the LUT such that

$$J1 \leq k \leq J2$$

J1 and J2 are input parameters which specify the anticipated lower and upper LUT interval limits over which the subsequent interpolations will be required.

When the routine has finished, an LUT value at (x,y) is given by

$$s_k x + b_k$$

where k is the LUT column interval in which lies.

4.77 SEARCH

Purpose

The SEARCH routine performs a CDC PFSUB function (GET, ATTACH, etc.) so that a file need not necessarily reside in the user's catalog when he runs CEL-1, provided it resides in the master CEL-1 catalog, M4800GS. In this manner, for example, another user may access and use photometric data which exists only in M4800GS.

Method

FUNC (1 word character string) specifies the function to be performed. This will usually be an attempt to GET or ATTACH a file. If possible, the function is performed on the current catalog; if this attempt fails, the function is attempted on the master catalog M4800GS. If both attempts fail, IW is set to -1 and control returns to the calling program.

4.78 SKYLUM

Purpose

This routine returns the luminance at a given point in the sky.

Method 4,9

ZL = zenith luminance
AZ = solar azimuth, relative to room north
EL = solar elevation
ZS = solar declination from zenith = 90-EL
PAZ = azimuth of the sky point, relative to room north
PEL = elevation of the sky point
PSI = angle subtended at the earth by the sun and the sky point

1. OVercast sky:

$$\text{SKYLUM} = \frac{ZL}{3}(1 + 2\sin PEL)$$

2. Clear sky:

$$\text{SKYLUM} = \frac{ZL\ (1-e^{-.32/\sin PEL})\ (.91 + 10e^{-3PSI} + .45\cos^2 PSI)}{.274\ (.91 + 10e^{-ZS} + .45\cos^2 ZS)}$$

3. Partly cloudy sky:

$$\text{SKYLUM} = \frac{ZL}{3}\left[\frac{.91 + 10e^{-3PSI} + .45\cos^2 PSI}{.91 + 10e^{-ZS} + .45\cos^2 ZS} + 2\sin PEL\right]$$

4.79 SOLPOS

Purpose

SOLPOS computes the azimuth and elevation of the sun as a function of day of the year and position on earth.

Method'

Let: L = latitude of the site
t = solar time
DAY = Julian day

$$U = 23.45 \sin \frac{360}{365}(\mathrm{DAY} - 80.75)$$

$$w1 = -\cos 15t \sin L \cos U - \cos L \sin U$$

$$w2 = \sin 15t \cos U$$

$$w3 = -\cos 15t \cos L \cos U + \sin L \sin U$$

then

$$\text{solar elevation} = \tan^{-1}\left[\frac{w3}{(w1^2 + w2^2)^{1/2}}\right]$$

$$\text{solar azimuth} = \begin{cases} \tan^{-1}(w2/w1) & \text{if } w1 \geq 0 \\ \tan^{-1}(w2/w1) + 180\,\mathrm{sig}(w2) & \text{if } w1 < 0 \end{cases}$$

$$\text{where } \mathrm{sig}(w2) = \begin{cases} 1 & \text{if } w2 > 0 \\ -1 & \text{if } w2 \leq 0 \end{cases}$$

4.80 SPHCON

Purpose

This routine computes sphere contrast, which is need for ESI calculations.

Method

Let B_{ij} and T_{ij} be the background and target luminance BRDF factors, respectively, for i=1, ... ,19 and j=1, ... ,37 (i corresponds to the 19 vertical angles 0, 5, ... ,90; j corresponds to the 37 lateral angles 0, 5, ... , 180)

Let:

$$BACK = \sum_{i=1}^{19} \sum_{j=1}^{37} w_i w_j \sin\theta_i \cos\theta_i B_{ij}$$

$$TASK = \sum_{i=1}^{19} \sum_{j=1}^{37} w_i w_j \sin\theta_i \cos\theta_i T_{ij}$$

$$w_i = \begin{cases} .5 & \text{if } i=1 \text{ or } i=19 \\ 1 & \text{otherwise} \end{cases}$$

$$w_j = \begin{cases} .5 & \text{if } j=1 \text{ or } j=37 \\ 1 & \text{otherwise} \end{cases}$$

$$\theta_i = 5(i-1) \text{ degrees}$$

Then

$$\text{sphere contrast} = \frac{|BACK-TASK|}{BACK}$$

4.81 SRSET

Purpose

SRSET returns the time of day (Local time) when the sun rises and sets.

Method

The "bisection" method is used. I.e., a function (solar altitude) is computed at each end and at the middle of a time interval. At each iteration one of the end points is replaced by the middle point (i.e., the interval is halved) and the new middle point becomes the average of the 2 "new" end points. The new interval at each step is chosen so that the function has opposite signs at each of the end points (i.e., the sun is below the horizon at one end and above the horizon at the other).

For example, to determine when the sun rises, we start with the interval (0,12) = (midnight,noon) and thus compute solar position at midnight, 6am, and noon. We find the sun below the horizon at midnight and at 6am, and above the horizon at noon -- hence our next interval becomes (6,12). We continue halving the interval in this fashion until the interval is shorter than 0.01 hour.

Note the formula for computing local time from solar:

local time = solar time - (CMER-LON) / 15 + DST + (equation of time)

where CMER = longitude at the center of the time zone
LON = longitude of the site
$$\text{DST} = \begin{cases} 0 \text{ if daylight savings time is not in effect} \\ 1 \text{ if daylight savings time is in effect} \end{cases}$$

4.82 STABLS (also ADDEM, WTABL)

Purpose

The output from STABLS is used by the LUMEN-II subset of programs in performing indirect calculations at the target points. STABLS computes LUTs which yield illumination/luminance at the target point due to a piece of wall or ceiling, one corner of which lies normal to the target point, with the position of the other corner being used to access the LUT.

Method

Let the target point be located at (0,0,0). Refer to Figure 4.82.a.

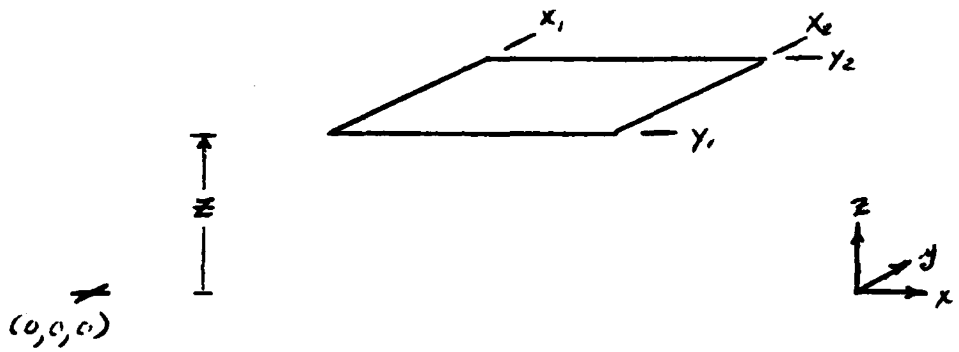


Figure 4.82.a: Illuminance at a point due to a lambertian area source in a plane parallel to the target plane.

The illuminance E at the target point due to the rectangular ceiling portion bounded by x_1, x_2 and y_1, y_2 is given by:

$$E = \frac{L}{2\pi} \left(F(x_1,y_2) - F(x_1,y_2) - F(x_2,y_1) + F(x_2,y_2) \right)$$

where

$$F(x,y) = \frac{x}{(x^2+z^2)^{1/2}} \tan^{-1}\frac{y}{(x^2+z^2)^{1/2}} + \frac{y}{(y^2+z^2)^{1/2}} \tan^{-1}\frac{x}{(y^2+z^2)^{1/2}}$$

Using this formula, we can compute the illuminance due to each LUT cell. Combining this with the BRDF values, tabular values of raw fc with body shadow, background luminance, and target luminance may also be computed. Subroutine ADDEM adds up these values to yield cumulative LUTs; i.e., the value at any LUT point represents the contribution to the target point from the entire rectangle whose opposite corners are located directly over the target point and at the LUT point in question.

The walls are handled in analogous fashion; refer to Figure 4.82.b:

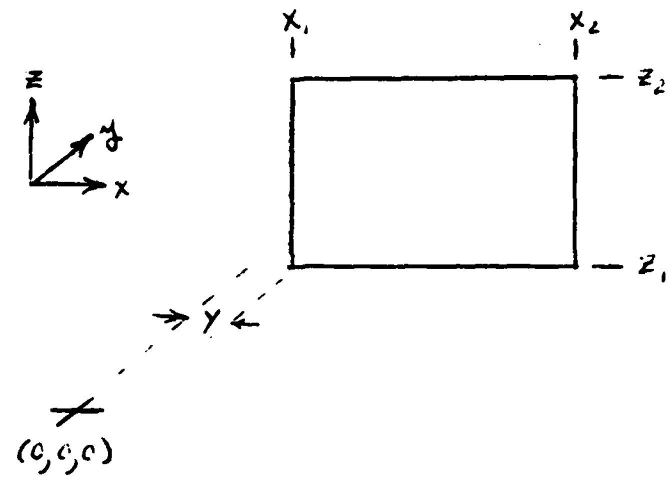


Figure 4.82.b: Illuminance at a point due to a lambertian area source in a plane perpendicular to the target plane.

The illuminance E at the target point due to the rectangular portion of the wall bounded by (x_1, x_2) and (z_1, z_2) is given by:

$$E = \frac{-Ly}{2\pi}\left(F(x_1,z_1) - F(x_1,z_2) - F(x_2,z_1) + F(x_2,z_2)\right)$$

where

$$F(x,z) = \frac{1}{(y^2+z^2)^{\frac{1}{2}}} \tan^{-1} \frac{x}{(y^2+z^2)^{\frac{1}{2}}}$$

Variables / Arrays

F - 21 x 21 work space for horizontal fc LUT
R - 21 x 21 work space for raw fc/ with body shadow LUT
B - 21 x 21 work space for background luminance LUT
T - 21 x 21 work space for target luminance LUT
LUC - logical unit # of disc file where ceiling LUTs are to be written
LUW - logical unit # of disc file where wall LUTs are to be written
SH - body shadow factors
BB - background luminance BRDF factors
BT - target luminance BRDF factors

4.83 STAP

Purpose

STAP computes statistics for illuminance values at the target points and area points. These are used to determine the luminaire gain settings for energy profile calculations.

Method

The coding is self-documenting.

4.84 STATIS (also SORT)

Purpose

STATIS computes statistical occurrence levels for UNKNOWN task locations (rectangular grid of target points). I.e., the routine tells what computed value divides the set at the following percentiles:

75%, 80%, 85%, 90%, 95%, 99%

Method

The grid of target points (T) is copied into a linear array (A) which is then sorted into descending order (subroutine SORT). When this is done, the x^{th} percentile dividing value is simply the element of A which is x% into A from the front.

Note that undefined calculated values (i.e., those exceeding 10^6) are not counted in this statistical summary.

Variables/ Arrays

T - input grid of calculation values
NR - # rows in T
NC - # columns in T
NV - # viewing directions
QTY - the calculated quantity (e.g., horizontal illumination)
A - the linear array into which T is copied prior to sorting
LU - logical unit # for printed output

4.85 STATP

Purpose

STATP is used in energy profile calculations. It prints a statistical summary of horizontal fc, ESI, and ESI Rating for a given daylight condition.

Method

The maximum, minimum, and average values are printed.

4.86 SUB1

Purpose

SUB1 prints the computed results on a rectangular grid (UNKNOWN task locations) where viewing direction applies. I.e., quantities will be ESI-related: ESI, Raw fc with body shadow, background luminance, target luminance, CRF, and LEF.

Method

The table is printed, along with maximum, minimum, average, and mean deviation. Details are discussed in the description of SUB2. Arrays and variables are the same except that for SUB1 the input array A may contain up to 4 planes in the third dimension, and NV and VIEW give the number of viewing directions and the viewing directions, respectively.

4.87 SUB2

Purpose

SUB2 is used to print rectangular grids of computed values for quantities not involving viewing directions -- <u>i.e.</u>, horizontal fc and room surface luminance.

Method

The rectangular grid of values is printed on ½" x ½" squares (assuming 6 lines per inch and 10 characters per inch). The following statistics are computed: average, maximum, minimum, mean deviation. Mean deviation is computed as follows:

Let m be the number of rows on the grid; let n be the number of columns; let a_{ij} represent each point on the grid. The mean M is given by:

$$M = \frac{1}{m \cdot n} \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}$$

The mean deviation is then

$$\text{Mean Deviation} = \frac{1}{m \cdot n} \sum_{i=1}^{m} \sum_{j=1}^{n} \left| M - a_{ij} \right|$$

Variables / Arrays

CW - carriage width indicator (assumed always = 2)
A - array of computed values to be printed
NR - # rows in the grid
NC - # columns in the grid
R - row coordinates
C - column coordinates
ID - 20 x 5 array containing 5 lines of label information
QTY - 32-character array identifying the quantity to be printed
HT - height of target points above floor (must be zero or less for room surface luminance)
ISURF - surface number (1-6) which grid of values lies on (=5 for target point grid
JUNIT - 1 for English units
2 for metric units
JP - 1 for illumination
2 for luminance

4.88 UKSUM

Purpose

UKSUM is used in a partitioned and/or daylight environment to compute and print computed illumination values for unknown task locations (i.e., a rectangular grid of target points). For daylighting, UKSUM is invoked only in analysis mode; only a statistical summary of the computed values is printed in profile mode.

Method

First, the row and column coordinates of the grid are computed and converted to meters if necessary. Then the routine loops thru the indicators (array INDC) and prints the quantities specified there. Before printing, the linear list of computed values must be converted into a rectangular array; this is done by calling the subroutines LNTOU (horiz fc only) and KNTOU (all quantities except horiz. fc). Horizontal fc and ESI values are written to disc for subsequent plotting (call to subroutine WPLTOD) if indicated.

Variables / Arrays

XTP - x-coordinates of target points (linear list)
YTP - y-coordinates of target points
VIEWD - viewing direction (degrees) at each target point
FC - horizontal fc at each target point
ESI - ESI at each target point
RFC - raw fc w/body shadow at each target point
LB - background luminance at each target point
LT - target luminance at each target point
R - y-coordinates of the rows after conversion to a rectangular grid
C - x-coordinates of the columns after conversion to a rectangular grid
A - work space for converting the linear list of values into a rectangular grid
B - work space for converting the linear list of values into a rectangular grid

4.89 UTOKTL

Purpose

This routine converts a rectangular grid of "unknown" task locations into a linear linked list of calculation points. This is done so that the same algorithm may be applied to known task locations as to unknown task locations.

Method

The coordinates of each point in the grid are determined. Each point is combined in turn with each viewing direction; the resulting combination is then added to the list.

Note that 2 separate list links are constructed:

1) Links which connect all points for a given viewing direction
2) Links which connect all points sharing the same (x,y) location, regardless of viewing direction.

Thus there will be as many lists of type 1) as there are viewing directions, and as many lists of type 2) as there are distinct target point locations.

Variables / Arrays

TORG - the (x,y,z) coordinates of the target point nearest the origin
NTR - # rows of target points
NTC - # columns of target points
DRT - the distance between rows of target points
DCT - the distance between columns of target points
NV - # viewing directions
VIEW - vector of viewing directions
HEADT - points to first cell of the list of target points based on target location
HEADV - points to first cell of the list of target points based on viewing direction
LINKT - link field of list of target points based on target location
LINKV - link field of list of target points based on viewing direction
VIEWD - viewing direction of the target points
XTP, YTP - x and y-coordinates of target point locations
NTP - # of target points
NTASKL - # of task locations (= number of distinct target locations)

4.90 VFUNC

Purpose

VFUNC determines the "view" function between two points in the room. This view function is a measure of whether or not a line from one point to the other intersects any partitions.

Method

The calling arguments to VFUNC include pointers to 4 "quadrant-wise" partition lists. Each of these is a list of partitions which could possibly obscure any part of the luminaire in the quadrant in question. Looking from the top, the quadrants are numbered as below:

2	1
3	4

For example, a partition surface which lies entirely southwest of the southwest luminaire corner could only provide interference in quadrant 3 and hence need not be present in any of the lists for quadrants 1, 2, or 4.

Let (x_L,y_L,z_L) be luminaire location, and let (x_p,y_p,z_p) be the location of the target point. The view function is determined by first choosing the quadrant whose partition list to search; this depends on the relationship of (x_L,y_L) to (x_p,y_p). For example, if $x_p \gtrsim x_L$ and $y_p \gtrsim y_L$ then the list to search is that for the first quadrant.

Each partition on the list is examined to see if it blocks the view from one point to the other. If at least one partition blocks the view, then the view function = 0; if none do, VF = 1. To see how the determination is made for each partition, suppose the partition in question is vertical and runs north-south. Its x-coordinate is x_s and its y and z-coordinate lower and upper limits are (y_{1s},y_{2s}) and (z_{1s},z_{2s}), respectively.

The equation giving the location of a point $\vec{q}$ moving along the line from $\vec{L}$ to $\vec{p}$ can be written

$$\vec{q} = \vec{L}(1-\lambda) + \vec{p} = \vec{L} + \lambda(\vec{p} - \vec{L})$$

To determine the value of λ which puts $\vec{q}$ on the partition surface, we solve for λ in

$$x = x_L + \lambda(x_p - x_L)$$

where x is the x-coordinate of the intersection at the partition surface.

Using the value of λ thus obtained, we compute the y and z coordinates where the line intersects the partition:

$$y = y_L + \lambda(y_p - y_L)$$

$$z = z_L + \lambda(z_p - z_L)$$

The partition interferes if and only if

$$y_{1s} \leq y \leq y_{2s} \quad \text{and} \quad z_{1s} \leq z \leq z_{2s}$$

<u>Variables / Arrays</u>

XL, YL, ZL - (x,y,z) coordinates of the luminaire location

XP, YP, ZP - (x,y,z) coordinates of the target point location

HEAD, X1, X2, Y1, Y2, Z1, Z2, LINK, DIR - define the list structure which describes the partitions

LAMBDA - the parameter used as discussed in "Method"

4.91 VSKY

Purpose

VSKY computes a multiplier which yields the illumination on a vertical surface due to the unobstructed sky. The multiplier computed by VSKY is applied to the horizontal illumination from the unobstructed sky to yield the vertical illumination.

Method

The multiplier (VSKY) is computed as follows:

a) overcast sky: VSKY = 0.396

b) Clear sky:

(This space intentionally left blank)

Let ϕ be the angle between a normal to the vertical surface and the projection of the solar azimuth onto the horizontal plane. Then

$$VSKY = \begin{cases} -.00214\,\alpha_s + .387 + \left(1 + \sin\left(180 + \frac{90}{110}|\phi|\right)\right)\left(-0.023\,\alpha_s + 1.92\right) & \text{if } |\phi| \le 110 \\ -.00214\,\alpha_s + .387 + \left(|\phi| - 110\right)\left(-0.0000194\,\alpha_s + .0012\right) & \text{if } |\phi| > 110 \end{cases}$$

c) partly cloudy sky:

$$VSKY = \begin{cases} 0.31 + \cos\phi\,(.322 - 0.00223\,\alpha_s) & \text{if } |\phi| \le 90 \\ 0.31 + \sin(|\phi| - 90)(.216 - .0046\,\alpha_s) & \text{if } |\phi| > 90 \end{cases}$$

α_s is the solar altitude, in degrees.

Variables / Arrays

NORM - (x,y,z) coordinates of a vector which is normal to the vertical surface (extends outward from the surface). Since the surface is vertical, NORM(3) is by definition zero and is therefore not used in the calculations here.

AZ - solar azimuth (degrees) relative to room north

EL - solar altitude (Degrees) above the horizon

ISKY - sky condition (1 = overcast, 2 = clear, 3 = partly cloudy)

AN - the angle $|\phi|$ as discussed in "Method"

4.92 WACCUM

Purpose

WACCUM is used in a non-partitioned, non-daylight environment to accumulate initial illuminance to an array of points on a wall from a line source which is parallel to the wall. The aggregate such contributions from all luminaires may be used in the flux transfer analysis to determine the final average luminance on the room surfaces.

Method

For each row of points on the wall (i.e., all points on the row have the same z-coordinate), the following is done:

1. The line source is descretized so that the maximum piece is no greater than ¼ the shortest distance from the source to the wall.

2. A piecewise linear fit is performed on the point source LUT. This fit gives the LUT value at a given row index into the table. I.e., if we desire a value from position (x,y) in the LUT, the value is given by

$$fc = s_k x + b_k$$

   where s_k and b_k are constants over the k^{th} interval in the LUT. The routine SANDB computes the (s_k, b_k), one pair for each LUT interval which can possible be touched by the most distant combination of calculation point and luminaire piece location.

3. The routine loops thru each discrete piece of luminaire.

4. The routine loops thru each calculation point on this row, accumulating the contribution from the current luminaire piece.

Steps 1 - 4 are repeated for each row of points on the wall.

Variables / Arrays

LEN - length of line source luminaire
PST - 21 x 21 point source LUT
WALL - array of illumination to be computed on the wall
Z - distance from wall to luminaire
XLUM - distance from center of luminaire to the vertical edge of the wall which is nearest the origin
YLUM - distance from floor to center of luminaire
DELR - distance between rows of points on the wall
DELC - distance between columns of points on the wall
NR - # rows of points on the wall
NC - # columns of points on the wall
NDIV - # discrete pieces luminaire is divided into
DEL - length of a discrete luminaire piece
S,B - 20 element vectors giving the slope and intercept, respectively, of the linear fit across the LUT intervals

4.93 WFUNC

Purpose

WFUNC is the "view" function which determines whether or not one point can "see" another point because of possible interference from partitions.

Method

The algorithm employed is to trace rays from one point to the other, determining whether each ray can pass through a partition or not. This routine is almost identical to VFUNC, the only difference being that VFUNC is passed 4 different partition lists and chooses only one to trace rays through. WFUNC, on the other hand, is passed only one list and hence does no selecting.

VFUNC is used for determining the view function from target points; WFUNC is used to determine the view function from points on room and partition surfaces. For more details on the algorithm and variables used refer to the documentation of the VFUNC routine.

4.94 WRFTM

Purpose

WRFTM computes and writes the coefficients of the flux transfer matrix to disc. The same matrix can be used repeatedly with different values of initial illuminance to yield final surface luminances under, say, differing daylight conditions.

Method

Calculation of the flux transfer matrix requires the capability to compute the form factor from any surface i to any other surface j. ("Surface" means a planar rectangular area which may be a room surface, an insert, a fenestration opening, or one face of an obstruction within the room) The form factor F_{ij} is the fraction of flux leaving surface i which is directly incident upon surface j.

The equation giving the final average luminance on room surface j can be written:

$$L_j = \rho_j (E_{0j} + (A_1/A_j)F_{1j}L_1 + (A_2/A_j)F_{2j}L_2 + \dots + (A_n/A_j)F_{nj}L_n \qquad (1)$$

where ρ_j = reflectance of surface j

E_{0j} = initial illumination on j (before any interreflections are considered)

A_k = area of surface k

F_{kj} = form factor from surface k to surface j

When each surface is planar, as in this case, $F_{ii} = 0$ for all i. Also we may make use of the reciprocity relation

$$A_i F_{ij} = A_j F_{ji}$$

to rewrite (1) as

$$\rho_j F_{j1} L_1 + \rho_j F_{j2} L_2 + \dots - L_j + \rho_j F_{jn} L_n = -\rho_j E_{0j}$$

Writing this as a matrix equation, we have

$$A \vec{L} = \vec{E}$$

where $\vec{L}$ is the vector of unknown final luminances
$\vec{E}$ is the vector whose j^{th} element is given by $-\rho_j E_{0j}$
A is the flux transfer matrix with

$$A_{ij} = -1 \quad \text{for } i = j$$
$$A_{ij} = \rho_i F_{ij} \quad \text{for } i = j$$

WRFTM computes the form factors as if no intervening surfaces (furniture, etc.) could be present. From each surface i, all factors F_{ij} for j = i, are computed; these are then proportionally adjusted so that their sum is 1.

Form factors are computed by the routines FFPAR (parallel surfaces) and FFPER (perpendicular surfaces). First the coordinates of the surfaces in question are transposed into the relationships required by FFPAR and FFPER. For these relationships, refer to the documentation for FFPAR and FFPER.

Note that form factors on only one side of the main diagonal need be calculated, since the remainder may be obtained from the reciprocity relationship.

Variables / Arrays

HEAD, LINK, REFL, C1, C2 - list structure describing the partitions

FTM - array to contain the flux transfer matrix coefficients (max 75 partition surfaces allowed). Each column (75 words) constitutes one disc record.

LUFTM - logical unit # of disc file to which the flux transfer matrix is written.

4.95 WPLTOD

Purpose

This routine writes a di record of values which may be subsequently read by the character contour plotting logic.

Method

The array B (492 words long) contains information used to generate one character contour plot. The array of values itself (21 x 21 = 441 words) is copies into B, followed by the row and column coordinates (21 words apiece), the quantity identifier (8 words), and the viewing direction (1 word).

Variables / Arrays

LUPLOT - if zero, means no plots are to be generated. If not zero, LUPLOT is the logical unit # of the disc file where the record is to be written.

NPLOTS - counts the number of plots to be generated

A - array of computed values from which plots are to be constructed

NR - # rows in A
NC - # columns in A

VIEW - the viewing direction (degrees)

QUAN - Hollerith string (32 characters) identifying the quantity in A (e.g., HORIZONTAL ILLUMINATION)

ROWS - y-coordinates of rows

COLS - x-coordinates of columns

B - 492-word vector which is written to disc

C - 21 x 21 array, equivalenced to B

4.96 WTOSRF

Purpose

WTOSRF computes the candela multiplier table from a given fenestration rectangle to a target surface (obstruction face, insert, or room surface). The candela multiplier table is a 21 x 21 table at the asymmetric LUT points; when each value in this table is multiplied by the corresponding fenestration luminance, the sum of these products gives the average illuminance on the target surface.

Method

The fenestration rectangle is discretized into pieces no greater than 5' in either dimension. The routine then calls CTSURF once for each fenestration piece. For each piece, CTSURF computes the multiplier table for the surface.

Variables / Arrays

WSURF - integer which identifies the surface where the fenestration lies (1, 2, 3, 4, or 6)

W1 - the (x,y,z) coordinates of the fenestration corner nearest the origin

W2 - the (x,y,z) coordinates of the fenestration corner furthest from the origin

NR - # rows of points on the target surface

NC - # columns of points on the target surface

ORG - (x,y,z) coordinates of the point on the target surface which is nearest the origin

DELR - vector which steps from one row to the next on the target surface.

DELC - vector which steps from one column to the next on the target surface

TSURF - the direction faced by the target surface

T - 21 x 21 table of multipliers from each fenestration piece. These are added together to form the output table R.

WPSQFT - the area, in square feet, of each window piece after discretization

4.97 ZENLUM

Purpose

ZENLUM computes the zenith luminance for given sky conditions.

Method

For the given sky condition, horizontal illuminance from the unobstructed sky is computed. The zenith luminance is then computed from this horizontal fc value. This calculation method is appropriate, since the use of predetermined sky luminance distribution formulae dictates that the zenith luminance and the horizontal illuminance from the unobstructed sky are not independent. Let E be the horizontal illuminance from the unobstructed sky, and let Z be a divisor which is obtained from the implied fixed relationship between E and the zenith luminance. For the overcast sky Z is independent of solar altitude and has the value

$$\frac{7}{9}$$

Thus for the overcast sky we have zenith luminance = $\frac{9}{7}$ E

For the clear and overcast skies, tables of Z are stored for solar altitudes from 0° to 90° in steps of 10°. An interpolated Z is obtained from these tables, based on the solar altitude. Then,

$$\text{zenith luminance} = E / Z$$

Variables / Arrays

ISKY - defines sky condition (1=overcast, 2=clear, 3=partly cloudy)

SUNEL - solar elevation above horizon, in degrees

ICOND - dummy parameter, not currently used. This is intended to be an indicator of atmospheric conditions in future enhancements.

Y - 10 x 3 array which contains the tabulated values of Z versus sky condition and solar altitude. Note that one column of the array is for the overcast sky and hence contains all entries = .7777778

4.98 ZONEPC

Purpose

ZONEPC computes multipliers which are used to compute the indirect component in a partitioned / daylight environment. From a given target point, 48 rays are projected into the upper hemisphere, at azimuth angles 15, 45, ... , 315 degrees, and vertical angles 15, 37.5, 52.5, and 75 degrees. Each of the 48 rays is traced until it strikes an interior surface. The illumination / luminance at the target point is then the sum of the contributions from each of the 48 zones surrounding the projected rays. The contribution from each of the 48 zones is equal to the luminance of the zone times a multiplier which depends on the size of the zone and its location relative to the target point.

Method

ZONEPC computes these multipliers: sets of 48 each for horizontal footcandles, raw fc with body shadow, background luminance, and target luminance. The latter 3 are computed at each of the 16 viewing directions 0, 22½, 45, ..., 337½ degrees. The 48 rays are chosen in such a manner that the multiplier for horizontal fc is 1/48 for each zone. The remaining multipliers are 1/48 times the body shadow digit and BRDF factor for the angle in question.

The "raw" BRDF data in array A are on $5^{\circ} \times 5^{\circ}$ angle spacing. The data in A are distilled into the arrays SH, BB, and BT, so that these latter arrays contain the BRDF data at the angles needed for the 48-ray multipliers. The routine then loops thru the 48 rays and constructs the output multipliers from SH, BB, and BT.

Variables / Arrays

LU - logical unit # of disc file where results are to be written

A - 19 x 37 x 3 array of body shadow and BRDF factors vs. spherical coordinates

SH - 4 x 73 intermediate body shadow factor table computed from A

BB - same as SH, only for background luminance BRDF factors

BT - same as SH, only for target luminance BRDF factors

BRDF - 4 x 73 x 3 array which contains SH, BB, and BT. I.e., BRDF(*,*,1) = SH, BRDF(*,*,2) = BB, BRDF(*,*,3) = BT
An EQUIVALENCE statement effecting these must be given in the calling program.

TSH - 4 x 12 x 16 array of output multipliers for raw fc with body shadow. TSH(i,j,k) = multiplier for ray at vertical angle θ_i,

azimuth $\emptyset_j$, viewing direction V_k, where θ_i = 15, 37½, 52½, [illegible] degrees for i=1,2,3,4, respectively,

$\emptyset_j$ = 15 + (j-1)30

V_k = 22½ (k-1)

TBB - same as TSH, except for background luminance BRDF factors

TBT - same as TSH, except for target luminance BRDF factors

T1 - same as TSH, except for horizontal fc

(Note that T1, TSH, TBB, and TBT must occupy contiguous areas of memory. This may be accomplished via appropriate EQUIVALENCE statements in the calling program)

4.99 LUMSHF

Purpose

LUMSHF determines the solar flux falling on a light shelf, for a given solar-generated horizontal illuminance component.

Method

The top of the light shelf is discretized into pieces of maximum dimension 1' in either direction. A ray trace is then performed from the center of each piece to see if sunlight falls on the piece. The product of the solar illuminance and the area of the piece gives the solar flux on the piece. The solar flux on the entire shelf is the sum of the solar "fluxes" on each piece.

Variables / Arrays

IMAP - # of the prototype fenestration entry for the fenestration in question

IWALL - wall surface # (1, 2, 3, or 4) where the fenestration lies

VS - direction cosines of a vector directed toward the sun

SLUMEN - solar flux on the light shelf (output)

SUNFC - horizontal illuminance due to direct sunlight

4.100 BARBIL

Purpose

BARBIL traces a light ray from an interior point to the sun to see if the ray is obstructed by either an exterior building surface or by a barrier outside a window. The routine is used to determine the solar illuminance on target points and on light shelves beneath windows.

Method

When the routine is entered, it will already have been determined that the ray passes through a fenestration opening; this opening is identified in the calling arguments. If the opening is a vertical opening in ceiling fenestration (e.g., a vertical face on a clerestory), BARBIL first checks to see if the ray hits the room ceiling before reaching the vertical fenestration opening.

Next, if the fenestration is a window with one or more barriers outside, the ray trace is performed on the 3 barriers above and to each side of the window.

Finally, the ray trace is performed on external building surfaces (subroutine RASTRK). The ray-tracing logic is analogous to that described under RASTRK.

Variables / Arrays

WSURF - surface # where fenestration lies
WC1 - (x,y,z) coordinates of fenestration opening corner which is nearest the origin
WC2 - (x,y,z) coordinates of fenestration opening corner which is furthest from the origin
NB - # barriers outside the window (0 thru 3)
B1 - (x,y,z) coordinates of barrier corner which is nearest origin
B2 - (x,y,z) coordinates of barrier corner which is furthest from the origin
KBCON - identifies the constant coordinate of each barrier
ICEIL - 1 if fenestration is in ceiling, 0 if not
CF1 - (x,y,z) coordinates of corner of ceiling opening which is nearest the origin
CF2 - (x,y,z) coordinates of corner of ceiling opening which is furthest from the origin
TP - (x,y,z) coordinates of the point from which ray is traced
FP - (x,y,z) coordinates of the point on the vertical fenestration opening which is intersected by the ray from TP to the sun.

4.101 FURNSH

Purpose

The function FURNSH traces a ray directed from a target point toward the sun. The ray is traced to see if it strikes any furniture within the room or a light shelf.

Method

The subroutine WFUNC is used to trace the ray for interference by furniture. If no furniture interferes, the light shelf (if any) is checked with logic which is decribed under subroutine RASTRK.

Variables / Arrays

TP - source of projected ray toward the sun
FP - point of intersection of projected ray at the fenestration
PARTS,X1,X2,Y1,Y2,Z1,Z2,DIR - characterize the interior obstruction surfaces
SH1 - (x,y,z) coordinates of corner of light shelf which is nearest the origin
SH2 - (x,y,z) coordinates of corner of light shelf which is furthest from the origin

4.102 THRUW

Purpose

Given a point in the room, the direction cosines of a vector toward the sun, and a fenestration opening, this function determines if a ray from the point toward the sun passes through the fenestration opening.

Method

The ray-tracing scheme is the same as that described under subroutine RASTRK.

Variables / Arrays

TP - (x,y,z) coordinates of the source of the projected ray

FP - (x,y,z) coordinates of the point of intersection of the projected ray with the fenestration (output from the function)

U - direction cosines of the unit vector directed toward the sun

WC1 - (x,y,z) coordinates of the corner of the fenestration opening which is nearest the origin

WC2 - (x,y,z) coordinates of the corner of the fenestration opening which is furthest from the origin

IW - the number of the surface the fenestration opening lies on.

4.103 TPSUN

Purpose

TPSUN determines whether or not points inside the room are exposed to direct sunlight through a given fenestration opening. The points for which the check is made include calculation target points, interior sensors, and control target area points.

Method

If a light shelf is present under the fenestration opening in question, its boundary (x,y,z) coordinates are computed. Then for each interior calculation point the following checks are made:

1. Does the ray from the target point to the sun pass through the fenestration opening? (function THRUW)

2. Does the ray strike the light shelf (if any) or any interior obstructions ? (function FURNSH)

3. Does the ray strike a window barrier (if any) or an external building surface? (function BARBIL)

SECTION V

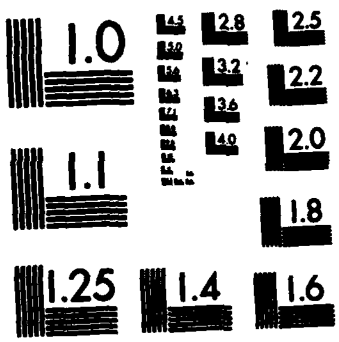
Logical Unit Assignments

5.1 Logical Unit Assignments

The tabulation below shows the logical unit numbers of the disc files used in CEL-1, together with the description and, where one exists, the symbolic variable name used to refer to the file:

#	symbol	description
5	-	input data deck
6	LUECHO	printed output
7	LU	data deck output from preprocessor
8	LUBAS	common block /COBAS/
9	LUCOB	common block /COBRDF/
10	LUINS	insert definitions
11	LU11	accumulated illuminance on target points and surfaces
12	LU12	point and area source LUTs
13	LULUM	luminaire locations and orientations
14	LUDB	obstruction database
15	LUZONE	zonal multipliers for 48-ray projections
16	LUFCTS	FCTs on interior surfaces, sensors, and area points
17	LUS48	48-ray destinations for exterior sensors
18	LUSTA	CLOUDS database file
19	LUDIM	parameters for luminaire control / energy profile
20	LUSEN	sensor definitions (Locations, etc.)
21	LUPAE	interior partition surface definitions
22	LUFTM	flux transfer matrix
23	PSLUT	point source LUTs (but not for LUMEN-II subset)
24	LUIOS	accumulated point-by-point illuminance on interior surfaces
25	LUCONT,LUQ	contribution from each luminaire to target points, sensors, area points.
26	LU26	VCP LUTs
27	LU27	accumulated VCP-related quantities
28	LUQ	quasi-luminaire contributions
29	LUKTL	defines linear array of target point coordinates, viewing directions, etc.
30	LUFEN	fenestration definitions
31	LUSURF	exterior building and ground surface definitions
32	LUDAY	latitude, longitude, occupancy, etc.
33	LUBRDF	BRDF LUTs

34	(not used)	
35	LUCEIL	LUTs for ceiling contribution
36	LUWALL	LUTs for wall contribution
37	LUB48	48-ray projection destinations from barriers
38	LUPLOT	records for character plots
39	LUFCTP	FCTs for target points
40	LUCMPR	"compressed" luminaire contributions
41	LUPRO	prototype FCTs
42	LUSLUM	relative initial illuminance (point-by-point) on room surfaces due to fenestration
43	LUSKED	schedules for BLAST interface
45	LUDEF	daylight effects for each time instant
46	LUSUN	direct solar component
48	LU48	destinations of 48-ray projections from the target points
61	-	body shadow factors
62	LUBK	background luminance BRDF factors
63	LUTA	target luminance BRDF factors
64	-	luminance values corresponding to RCS values
72	LUER	ERRORS file (used by data checker)


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5

SECTION VI

Compiling the Programs

6.1 FORT Procedure

The FORT procedure is virtually identical to the NOS-supplied FTNN procedure file for interactive compilation of FORTRAN programs. The only difference is that FORT performs no sort operations; FTNN performs a sort which renders it incapable of handling standard FORTRAN card format. In all other respects, FORT and FTNN appear and behave in identical fashion.

FORT takes its input as traditional FORTRAN card images; however, the first card in any deck to be compiled must be

```
00100$FMT,0
```

This card tells FORT that the remainder of the deck is standard FORTRAN card-image format which contains no line numbers.

To invoke the procedure simply key in

```
-FORT(A=name)
```

where name is the name of a local file which contains the FORTRAN source statements. Compiler options may then be entered interactively, as with FTNN. For full details, refer to the discussion of FTNN in NOS Computing Services Reference Set, Volume 2, pp. 7-22 thru 7-31.

6.2 Main Program Source Files

A main program source file contains a main program plus zero or more subprograms. E.g., to compile and replace the existing binary relocatable version of CEL01:

```
GET,ZCEL01

-FORT(A=ZCEL01)

FTN OPTIONS ?  B=CEL01,C

REPLACE,CEL01
```

If no binary version of CEL01 exists prior to the compilation, the last statement would be

```
SAVE,CEL01
```

6.3 Subprogram Source Files

A subprogram source file contains only subroutines and functions -- no main programs. The CEL-1 scheme calls for storing the relocatable binary code from all subprogram source files into the library file RELOC.

The CEL-1 procedure files (CELPROC, OBMPROC, CCMPROC) instruct the NOS loader to resolve subprogram references from RELOC, so it is imperative that the binary code for separately-compiled subprograms reside in RELOC.

For example, to compile the subprogram source file ZDISECT and place the relocatable binary code into RELOC:

```
GET,ZDISECT
-FORT(A=ZDISECT)
FTN OPTIONS ?  M=RELOC
```

SECTION VII

FORTRAN Source Files
and
Other Auxiliary Files

7.1 Main Program Source Files

The following list shows the FORTRAN main program source files, together with the main program each contains, plus any subprograms contained in the same file. In all cases, the first routine name following the file name is the name of the main program.

Filename	Routines
ZCCMP	CCMP
ZCELIST	CELIST TOF
ZCELKIL	CELKILL
ZCEL01	CEL01 SCOMP
ZCEL02	CEL02
ZCEL031	CEL031
ZCEL032	CEL032
ZCEL033	CEL033
ZCEL034	CEL034 TRIGS
ZCEL035	CEL035 TRIGS
ZCEL04	CEL04
ZCEL07	CEL07 ENGMET
ZCEL1PP	CEL1PP KEY
ZCHECK	CHECK
ZIFACE	IFACE
ZOBMP	OBMP
ZOVLY20	OVLY20
ZOVLY30	OVLY30 ILAVG INVRS RFAVG FLUM PERFF
ZOVLY40	ENORM EPARL ADWAL ADCEIL
ZOVLY50	CALVCP INTSRF
ZOVLY60	OVLY60
ZSUNHIT	SUNHIT TPLOOP RAYSFW ADJFP IRASUN CSUN QUART RANGE
ZCELSOL	CELSOL

7.2 Subprogram Source Files

The following list shows the FORTRAN subprogram source files, together with the names of the subroutines and functions contained in each one.

Filename	Routines
ZACCAS	ACCAS MAPAR IIOS
ZACC1L	ACC1L ESIRAT
ZADBAR	ADBAR
ZADDRS	ADDRS
ZADJFL	ADJFL
ZADSORT	ADSORT
ZAPART	APART
ZBARFL	BARFL
ZBARVAL	BARVAL
ZBASET	BASET VFBLA
ZBGRAH	BGRAH
ZBILFSL	BILFSL
ZBILPAR	BILPAR OBSURF INCEL GETCEL RETCEL
ZBILTAS	BILTAS
ZBLILUM	BLILUM FFBL
ZBLSKYV	BLSKYV
ZBPART	BPART
ZBRDF	BRDF
ZCANDMT	CANDMT
ZCHROOM	CHROOM
ZCLUT	CLUT PRJECT
ZCMPRES	CMPRES
ZCMTFES	CMTFES
ZCONMAP	CONMAP
ZCONTR1	CONTR1
ZCPARTL	CPARTL
ZCPR048	CPR048 RAY6
ZCTIMES	CTIMES
ZCTSURF	CTSURF
ZCVMET	CVMET
ZDAYEF	DAYEF

Filename	Routines
ZDIGITZ	DIGITZ
ZDISECT	DISECT GFORM FLUSH ICH FORM ER OBDB CCDB
ZD048	D048
ZDR48	DR48
ZEQTIME	EQTIME
ZES48	ES48
ZFCBS	FCBS
ZFCESEN	FCESEN
ZFCTDSK	FCTDSK
ZFCTTP	FCTTP
ZFCTWTP	FCTWTP
ZFENGET	FENGET
ZFESI	FESI
ZFFPAR	FFPAR
ZFFPER	FFPER
ZFILPRT	FILPRT
ZFSTOTP	FSTOTP DISCW
ZFTAPAR	FTAPAR
ZFTPDSK	FTPDSK
ZGSEIDL	GSEIDL
ZGSET	GSET
ZHFCS	HFCSUN HFCSKY
ZINDTPS	INDTPS
ZIOERR	IOERR
ZJULDAY	JULDAY
ZKNTOU	KNTOU LNTOU
ZLGAINS	LGAINS
ZMETENG	METENG
ZMSTRSN	MSTRSN
ZOBSF	OBSF
ZOPTMZ	OPTMZ GRADE

Filename	Routines
ZOSCU	OSCU CEVAL SLOPE
ZOVLY21	OVLY21
ZOVLY22	OVLY22
ZOVLY23	OVLY23 AREALT INDIR
ZOVLY24	OVLY24 VCPTBL
ZOVLY25	ACCUM ACCBS ACCVCP
ZPHOTO	PHOTO
ZPLOTS	PLOTS PVALS PLUG TCELLS CONTUR PSAL CUPOL
ZPROFCT	PROFCT
ZPROFIL	PROFIL
ZPROJ48	PROJ48
ZQLUM	QLUM
ZRASTRK	RASTRK
ZRCSBP	RCSBP
ZRELFC	RSDET ICOVER RELFC VWTOSP HSTOSP
ZRHOEF	RHOEF
ZRSLTS	RSLTS
ZSANDB	SANDB
ZSEARCH	SEARCH
ZSKYLUM	SKYLUM
ZSOLPOS	SOLPOS
ZSPHCON	SPHCON
ZSPHERE	SPHERE
ZSRSET	SRSET
ZSTABLS	STABLS WTABL ADDEM
ZSTAP	STAP
ZSTATIS	STATIS SORT
ZSTATP	STATP
ZSUB1	SUB1
ZSUB2	SUB2
ZUKSUM	UKSUM
ZUTOKTL	UTOKTL
ZVFUNC	VFUNC

Filename	Routines
ZVSKY	VSKY
ZWACCUM	WACCUM
ZWFUNC	WFUNC
ZWPLTOD	WPLTOD
ZWRFTM	WRFTM
ZWTOSRF	WTOSRF
ZZENLUM	ZENLUM
ZZONEPC	ZONEPC
ZBARR3	BARR3
ZBARBIL	BARBIL
ZFURNSH	FURNSH
ZLUMSHF	LUMSHF
ZTHRUW	THRUW
ZTPSUN	TPSUN

7.3 Photometric Files

Photometric files are stores as HBxx, where xx is the 1 or 2 digit number identifying the luminaire according to Figure 9-12 of the IES Handbook (6th edition, Reference Volume). Photometric data for luminaires number 1 throuth 49 (HB1 thru HB49) are saved.

7.4 BRDF and Body Shadow Files

These files contain BRDF factors stored on a $5^{o} \times 5^{o}$ spherical grid. If a filename has last character B, the file contains background luminance BRDF factors; if the filename has last character T, the file contains target luminance BRDF factors. Files are:

B25B	B25T	P325B	P325T
D25B	D25T	P40B	P40T
F25B	F25T	P55B	P55T
O25B	O25T	T25B	T25T
P10B	P10T	X25B	X25T
P25B	P25T		

The file SHADOW contains the IES body shadow factors.

7.5 Other Auxiliary Files

RCS866 contains tabulated luminance for uniformly-spaced RCS values

ERRORS contains the error messages used in program CHECK

SITES contains the addresses and remote batch terminal logon codes for each NAVFAC site. Used by program IFACE.

OBSTR the obstruction database

CLOUDS the cloudiness database

RELOC contains relocatable binary code for all separately-compiled subprogram files.

7.6 Procedure Files

CELPROC executes CEL1

OBMPROC executes OBMP

CCMPROC executes CCMP

OUTLIST lists output files

CEL1FE executes the preprocessor

CEL1II permits user to submit CEL-1 jobs interactively

References

1. "A Method of Evaluating the Visual Effectiveness of Lighting Systems", RQQ Report No. 4, Illuminating Engineering, Vol. 65, August 1970, p. 504.

2. "Outline of a Standard Procedure for Computing Visual Comfort Ratings for Interior Lighting", RQQ Report No. 2, Illuminating Engineering, Vol. 61, October 1966, p. 643.

3. "The Predetermination of Contrast Rendition Factors for the Calculation of Equivalent Sphere Illumination", RQQ Report No. 5, Journal of the IES, January 1973, p. 149.

4. "Standardization of Luminance Distribution on Clear Skies", Publication CIE No. 22 (TC-4.2.) 1973.

5. D. DiLaura, "On the Computation of VCP", Journal of the IES, Vol. 5, no. 4, July 1976, p. 207.

6. R. Levin, "Position Index in VCP Calculations", Journal of the IES, January 1975, p. 99

7. P. O'Brien, "Effective Reflectance of Room Cavities with Specular and Diffuse Surfaces," Illuminating Engineering, April 1966, p. 189.

8. G. Gillette, W. Pierpoint, and S. Treado, "A General Illuminance Model for Daylight Availability", paper to be presented at the 1982 IES National Technical Conference.

9. W. Pierpoint, "Sun and Sky Models for Daylighting Design", program # Z0362-01-211C, Naval Civil Engineering Laboratory (Port Hueneme, Calif.).

APPENDIX A - COMMON Block Variable Definitions

A-1 Common Block /COBAS/

This common block is the "basic" block in that it contains variables used by virtually all the programs.

ROOMD - the x, y, and z dimensions of the room
TORG - the (x,y,z) coordinates of the point on a rectangular grid of target points which is nearest the origin
VIEW - the viewing directions for a rectangular grid of target points
RHO - the reflectance of the six room surfaces
WATTS - the watts consumption per luminaire at 100% gain
SQUAD - quadratic coefficients giving watts vs. gain
GMIN - minimum gain to which a luminaire can be dimmed
LORG - the point on a rectangular grid of luminaires which is nearest the origin (used only in design synthesizer)
NDIV - # of discrete zones into which the room surfaces are divided - in order, x, y, and z directions
INDC - 15 binary indicators governing which quantities are to be computed/printed:

1 - horizontal illumination
2 - ESI
3 - raw illum. w/body shadow
4 - LEF
5 - CRF
6 - target luminance
7 - background luminance
8 - VCP
9 - plot horizontal illum.
10 - plot ESI
11 - plot VCP
12 - Sun trace in room
13 - BLAST interface output
14 - room surface luminances
15 - not used

DESC - 5 lines of descriptive text to label output
DOESI - (0 = don't compute ESI, 1 = do compute ESI) Equivalenced to INDC(2)
OUNITS - output units (1=English, 2=metric)
IUNITS - input units (1=English, 2=metric)
IKNOWN - 1 = ESI Ratings to be computed at task locations
2 = UNKNOWN task locations (rectangular grid)
3 = KNOWN task locations
NTC - # columns of target points (UNKNOWN task locations)
NTR - # rows of target points (UNKNOWN task locations)
DCT - grid column spacing (UNKNOWN task locations)
DRT - grid row spacing (UNKNOWN task locations)
NV - # viewing directions
NFTYPE - # fenestration sub-blocks actually describing fenestration
NWIND - # fenestration source elements (windows, skylights, etc.)
NSURF - # exterior building surfaces + # ground surfaces
NTASKL - # task locations
NTP - # target points
NFOTO - # different photometric files used
GREF - reflectance of ground
NINS - # of room surface inserts
IOP - 1 ESI Rating at IES viewing directions
2 ESI Rating at Navy viewing directions
NEWP -=NDIV(1) = # discrete room zones in x-direction

NNSP   - =NDIV(2) = # discrete room zones in y-direction
NUDP   - =NDIV(3) = # discrete room zones in z-direction
ROOMAZ - azimuth of west wall of room, relative to room north
NLUM   - = NLUMC = # luminaires
SUNAZ - azimuth of sun, relative to room north
SUNEL - elevation of sun
SINAZ - sine(SUNAZ)
COSAS - cosine(SUNAZ)
SINEL - sine(SUNEL)
COSEL - cosine(SUNEL)
TANEL - tangent(SUNEL)
DMILL - minimum illuminance criterion for design synthesis
DAILL - average illuminance criterion for design synthesis
DMESI - minimum ESI criterion for desing synthesis
IPROF - (1= energy profile, 0=daylight analysis mode, -1=no daylight)
IDESGN - (1=design synthesis mode, 0=not design synthesis mode)
NLUMC - # columns of luminaires in rectangular grid
NLUMR - # rows of luminaires in rectangular grid
DELUMC - column spacing in rectangular grid of luminaires
DELUMR - row spacing in rectangular grid of luminaires
EYE   - observer eye height for VCP calculations
NPLOTS - # character contour plots to draw
NSNIN - # interior sensors
NSNEX - # exterior sensors
NAPTS - # area points (control target area)
NMAP   - # fenestration map entries

A-2 Common Block /FENREC/

This block contains variables which describe fenestration (see section 2.42 for more details on some of these variables).

FTYPE - type of fenestration (1=window, 2=clerestory, 3=sawtooth, 4=skylight)
FTRANS- transmittance of each face of fenestration
FREF - reflectance of each face of fenestration
FDIM - dimensions of the fenestration structures
BDIS - barrier "distance"
BL - barrier "Limit 1" and "limit 2"
BH - barrier "protrusion"
BREF - barrier reflectance
SHADET - shade transmittance
SHADED - shade pull-down depth
DRAPET - drape transmittance
DRAPED - drape close distance
BLTYPE - blinds type (1=horizontal, 2=vertical)
BLTHK - blinds thickness
BLW - width of one blinds vane
BLS - spacing between blinds vanes
BLANG - blinds angle setting
BLREF - blinds reflectance
SHZ - light shelf distance beneath bottom of window
SHY - light shelf protrusion distance from wall
SHR - light shelf reflectance
WC1 - coordinates of fenestration source element which are nearest the origin
WC2 - coordinates of fenestration source element which are furthest from the origin
FSHR - glazing parameter (1=clear, 2=diffusing)
NFLOC - # fenestration source elements per sub-block
PROSRF - surface # of fenestration prototype entry
PROTYP - pointer to FTYPE describing fenestration prototype entry
WPRO - pointer to PROTYP entry for each fenestration source element
MAPFT - pointer to FTYPE for fenestration map entry
MAPSU - surface # of each fenestration map entry
MAPGL - glazing parameter for each fenestration map entry

A-3 Common Block /CDIM/

This block contains some parameters used in energy profile calculations.

APX,APY	-	(x,y) coordinates of area points (control target area)
TORGAP	-	(x,y) coordinates of the point on the area points grid which is nearest the origin
DBAND	-	the criterion illuminance values for energy profile calculations
MINTP	-	minimum illuminance requirement on the target points
AVGTP	-	average illuminance requirement on the target points
MINAP	-	minimum illuminance requirement on the area points
AVGAP	-	average illuminance requirement on the area points
MINESI	-	minimum ESI requirement on the target points
CMETH	-	1 on-off controls 2 high-low-off controls 3 continuous dimming
LGRP	-	array which identifies the dimming group each luminaire belongs to. All luminaires which are controlled alike (<u>i.e.</u>, must have the same gain) belong in one dimming group.

A-4 Common Block /COMN1/

This block describes the partition surfaces within the room. More details may be found in Sections 2.3 and 4.13, so some of the variables in the block are omitted from the description here.

PARTS - 0 if no obstructions are present in the room
1 if at least one obstruction is present in the room

ICELNG - the surface # of the ceiling in the list of interior partition surfaces

IFLOOR - the surface # of the floor in the list of interior partition surfaces

NPSURF - the number of interior surfaces

A-5 Common Block /COBRDF/

This block is used in the Electric Light subset only:

ALUM - average final luminance on each room surface
AREF - average reflectance on each room surface
AILL - average illuminance on each room surface
QHO - reflectance of each room surface
INSRF - tells which room surface each insert lies on
LIMIN - (x,y,z) coordinate limits of each insert
RHOIN - reflectance of each insert
A - 3 x 3 rotation matrix which describes the luminaire orientation
LPOS - (x,y) coordinates of luminaire
LPROJ - projected (x,y) dimensions of luminaire
ORD - LUT ordinates (see section 2.1)
INDX - LUT index pointers (see section 2.1)
SH - body shadow factors
BB - background luminance BRDF factors
BT - task luminance BRDF factors
XEW - x-dimension of luminaire
YNS - y-dimension of luminaire
HLUM - z-dimension of luminaire
KUD - 1 if luminaire is downlight only
2 if luminaire is uplight only
3 if luminaire is both uplight and downlight
ZETA - measures offset of room nadir from luminaire photometric nadir
ALPHA, BETA, GAMMA - bearing, tilt, cant of luminaire
LASTA, LASTB, LASTG - bearing, tilt, cant of previous luminaire
TPARM - controls degree of luminaire discretization
SPHERE - sphere contrast
LTOS - accumulates "bright spot" luminance to subtract from second and later bounce indirect component calculations
ZL - z-coordinate of luminaire
JQCAL - (0 = don't compute ESI, 1 = do compute ESI)

A-6 Common Block /COM20/

This block is used in the Electric Light subset only:

VNDX - vertical angle index array (see sec. 4.65)
LNDX - lateral angle index array (see sec. 4.65)
CD - candela table
THETA - vertical angles
PHI - lateral angles
PT - 21 x 21 LUT work space
AL - stores luminance values of a "bright spot" (see sec. 3.12)
AI - work space for "bright spot" calculations
S,B - slope, intercept for interpolation between 2 LUT rows (see section 4.76)
NEWZ - # bright spot zones running east-west
NNSZ - # bright spot zones running north-south
XSPAN - ½ the total x-span of a bright spot
YSPAN - ½ the total y-span of a bright spot
RLUM - user-specified lamp lumens
RLLF - user-specified light loss factor
NPHI - # lateral angles in candela table
NTHETA - # vertical angles in candela table
KSYM - degree of symmetry in luminaire distribution (see sec. 4.65)

A-7 Common block /COCOMP/

This block is used when "compressing" luminaire contributions and in subsequent retrieval of the "compressed" records (see Section 4.22)

RCOMPR - vector which comprises the largest possible disc record. Contributions from more than one luminaire are packed into RCOMPR.
NLPR - # luminaires per disc record
NWPR - # words per record
NWPL - # words per luminaire (= NWPR / NLPR)
INRECD - indicates which disc record is currently in memory
LUCMPR - logical unit # where compressed records are stored
CALESI - (1=compute ESI, 0=don't compute ESI)
NTAR - # target points
NINSEN - # interior sensors
NCTAPS - # area points

APPENDIX B

Program Calling Sequence Trees

The sequences below define the subroutine and function linkages of the CEL-1 package. The notation is such that the "tree"

```
----------
A
  B
    C
  D
  E
    F
      G
      H
```

means that main program A directly calls subroutines B, D, and E. B directly calls C, E directly calls F, and F calls G and H. C, D, G, and H call no subroutines or functions.

```
1. ----------
   CHECK
     FLUSH
       DISECT
         GFORM
         ICH
     DISECT
       GFORM
       ICH
     ER
       FORM
         GFORM
     CCDB
       SEARCH
     OBDB
       SEARCH

2. ----------
   CEL01
     SCOMP
     METENG
     FENGET
       METENG
     BILFSL
     BILPAR
       INCEL
       SEARCH
       OBSURF
         GETCEL
       GETCEL
```

```
    CEL01 (continued)
      INCEL
      BILTAS
        GETCEL
      SEARCH
      SPHCON
      BRDF
      ZONEPC
      STABLS
        ADDEM
        WTABL

3.  ----------
    CEL02
      UTOKTL
      D048
        PROJ48
          ADSORT
      SEARCH
      PHOTO
      CLUT
      PRJECT
      ACCAS
        APART
        MAPAR
          IIOS
      CPARTL
      VFUNC
      FTAPAR
        GSEIDL
      ADJFL
      INDTPS

4.  ----------
    CEL031
      CONMAP
      VFBLA
        SOLPOS
        FCBS
          VSKY
        BASET
      FCTDSK
        CMTFES
          WTOSRF
            CTSURF
              APART
              CANDMT
                WFUNC
      RSDET
        APART
        VWTOSP
          RELFC
            WFUNC
        HSTOSP
          RELFC
            WFUNC
        ICOVER
```

```
5. ----------
   CEL032
     FTPDSK
       FSTOTP
         DISCW
           FCTWTP
             APART
             FCTTP
               CANDMT
                 WFUNC
               BLSKYV

6. ----------
   CEL033
     PROFCT
       CPRO48
         RAY6
         RASTRK
         FFPER
         FFPAR
       ADDRS
       DR48
         ADBAR
         BARVAL
           FFPAR
           FFPER
         RASTRK
       ADBAR
       RASTRK

7. ----------
   CEL034
     ES48
       ADBAR
       RASTRK
     CTIMES
     DAYEF
       TRIGS
       HFCSUN
       HFCSKY
       ZENLUM
         HFCSKY
       FCBS
         VSKY
       FILPRT
         BARFL
           RASTRK
           SKYLUM
           GSEIDL
         SKYLUM
         RASTRK
         BLILUM
           SKYLUM
           FFBL
             FFPAR
```

```
DAYEF (continued)
  GSEIDL
  ADJFL
  FCESEN
    BARFL
      RASTRK
      SKYLUM
      GSEIDL
    SKYLUM
    RASTRK
ZENLUM
  HFCSKY
SOLPOS
TRIGS
```

8. ----------

```
CEL035
  SEARCH
  CMPRES
  CTIMES
  MSTRSN
    ACC1L
      CONTR1
    ESIRAT
    CONTR1
    STAP
    GSET
    LGAINS
    FESI
      RCS
  SOLPOS
  HFCSUN
  HFCSKY
  FCBS
    VSKY
  RSLTS
    CVMET
    STATP
    UKSUM
      SUB2
      LNTOU
      WPLTOD
      KNTOU
      SUB1
  OSCU
    SLOPE
  CEVAL
  DIGITZ
  PROFIL
    BGRAH
  TRIGS
```

```
9. ----------
   SUNHIT
     CSUN
       QUART
         SRSET
         SOLPOS
         RANGE
     CHROOM
     APART
     BARR3
     TPLOOP
       RAYSFW
         ADJFP
         RASTRK
         IRASUN
           WFUNC

10. ----------
   CEL07
     ENGMET

11. ----------
   CEL04
     CMPRES
     OPTMZ
       GRADE
         FESI
           RCS
         ESIRAT
       ACC1L
     RSLTS
       CVMET
       STATP
       UKSUM
         SUB2
         LNTOU
         WPLTOD
         KNTOU
         SUB1

12. ----------
   OVLY20
     OBSF
     OVLY21
       SEARCH
       PHOTO
     OVLY22
       CLUT
       PRJECT
     WACCUM
       SANDB
     OVLY23
       AREALT
       INDIR
         SANDB
```

```
    OVLY20 (continued)
      OVLY24
        VCPTBL
      ACCUM
        SANDB
      ACCBS
        SANDB
      ACCVCP
        SANDB

13. ----------
    OVLY30
      ILAVG
      RFAVG
      FLUM
        PERFF
        INVRS

14. ----------
    OVLY40
      ENORM
      EPARL
      ADWAL
      ADCEIL

15. ----------
    OVLY50
      SUB2
      STATIS
        SORT
      FESI
        RCS
      SUB1
      WPLTOD
      CALVCP
      INTSRF

16. ----------
    OVLY60
      PLOTS
        PVALS
        CHROOM
        CONTUR
          TCELLS
            JSIGN
          PSAL
          OSCU
            SLOPE
          CUPOL

17. ----------
    CCMP
      SEARCH
```

```
18. ----------
   OBMP
     SEARCH
     IOERR

19. ----------
   CEL1PP
     SEARCH
     KEY
     CCDB
       SEARCH
     OBDB
       SEARCH

20. ----------
   IFACE
     (none)

21. ----------
   CELIST
     TOF

22. ----------
   CELSOL
     SOLPOS
     BPART
     TPSUN
       THRUW
       FURNSH
         WFUNC
       BARBIL
         RASTRK
       LUMSHF
         BARR3
         THRUW
         BARBIL
```

END

FILMED

3-83

DTIC